Nicholas Bush

Math 50

Section 1

HW 7

In problem 4 the only part I edited was play turn and so that was the only file I attached although I did include all of the other parts below.

Final hw problem 1:

you must have the fuka image in matlab somewhere to run this.

j = imread('fuka.jpg');

I = rgb2gray(j);

[x,y] = size(I);

myImage = imnoise(I,'salt & pepper', 0.02);

%i couldn't figure out why the edge needed four points and the sides needed

%5 so i put 6 points to predict the sides.

i=2;

B=zeros(size(myImage));

while(i<(x))

k=2;

while (k <(y))

window=zeros(9,1);

inc = 1;

for g=1:3

for h=1:3

window(inc)=myImage(i+g-2,k+h-2);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=med(5);

k=k+1;

end

i=i+1;

end

%find for the outer edges...

i=1;

k=2;

while (k<y)

window = zeros(6,1);

inc=1;

for g=1:2

for h=1:3

window(inc)=myImage(g,k+h-2);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(3)+med(4))/2;

k=k+1;

end

i=x;

k=2;

while(k<y)

inc=1;

window = zeros(6,1);

for g=i-1:i

for h=1:3

window(inc)=myImage(g,k+h-2);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(3)+med(4))/2;

k=k+1;

end

k=1;

i=2;

while (i<x)

window = zeros(6,1);

inc=1;

for g=1:3

for h=1:2

window(inc)=myImage(g+i-2,h);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(3)+med(4))/2;

i=i+1;

end

k=y;

i=2;

while (i<x)

window = zeros(6,1);

inc=1;

for g=1:3

for h=k-1:k

window(inc)=myImage(g+i-2,h);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(3)+med(4))/2;

i=i+1;

end

%for the courners...

i=1;

k=1;

while (i==1)

window = zeros(4,1);

inc=1;

for g=1:2

for h=1:2

window(inc)=myImage(g,h);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(2)+med(3))/2;

i=i+1;

end

i=1;

k=y;

while (i==1)

inc=1;

window = zeros(4,1);

for g=1:2

for h=k-1:k

window(inc)=myImage(g,h);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(2)+med(3))/2;

i=i+1;

end

i=x;

k=1;

while (i==x)

inc=1;

window = zeros(4,1);

for g=x-1:x

for h=1:2

window(inc)=myImage(g,h);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(2)+med(3))/2;

i=i+1;

end

i=x;

k=y;

while (i==x)

inc=1;

window = zeros(4,1);

for g=x-1:x

for h=y-1:y

window(inc)=myImage(g,h);

inc=inc+1;

end

end

med=sort(window);

B(i,k)=(med(2)+med(3))/2;

i=i+1;

end

B=uint8(B);

figure(1)

subplot(1,3,1)

imshow(I);

title('original image');

subplot(1,3,2)

imshow(myImage);

title('IMAGE WITH SALT AND PEPPER NOISE');

subplot(1,3,3)

imshow(B);

title('IMAGE AFTER MEDIAN FILTERING');

this code basically requires 9 different loops, (can be done with 5 loops and 4 other things, but I used 9 loops) the loops should be as follows, one loop for each corner, one loop for each side, and one loop for pretty much everywhere else. Each of the corner and side loops will be different as they each require different dimensions around them for instance the corner in the lower left hand side will require one square above one below its self and one to the left however for the lower right hand corner, if we attempt to take a value for a point t its left we will get an error.

Also I used salt and pepper noise, mostly because I didn’t want to scour the internet for a blurry image and also with this I can compare it to the original black and white image too.

Hw 7 problem 2:

n=150;

A=zeros(n,n);

A(:, 1) = 1;

j=2;

k=2;

while (j<=n)

k=2;

while (k<=j)

A((j),(k)) =(A((j-1),(k-1)) +A((j-1),(k)));

k=k+1;

end

j=j+1;

B = rem(A,2);

spy(B)

pause(.05)%should now run for 7.5 seconds

end

%this is pretty much the same code as number 4 in hw 4 but with a pause

%command in the middle and the spy command is in the while loop now.

Basically I took my problem 4 from homework 4 and placed a pause and spy command inside the loop so that it would pause for brief amounts of time inside the loop and we could see the rows as they were being made.

Hw 7 Problem 3

n=100;

A=zeros(n,n);

A(:, 1) = 1;

j=2;

k=2;

while (j<=n)

k=2;

while (k<=j)

A((j),(k)) =(A((j-1),(k-1)) +A((j-1),(k)));

k=k+1;

end

j=j+1;

end

B=rem(A,10);

spy(B)

pause(1)

C=rem(A,9);

spy(C)

pause(1)

D=rem(A,8);

spy(D)

pause(1)

E=rem(A,7);

spy(E)

pause(1)

F=rem(A,6);

spy(F)

pause(1)

G=rem(A,5);

spy(G)

pause(1)

H=rem(A,4);

spy(H)

pause(1)

I=rem(A,3);

spy(I)

pause(1)

J=rem(A,2);

spy(J)

pause(1)

%this script is similar to problem two however this time instead f changing

%the graph to display each row separately, I have changed the graph so that

%it will graph the remainder with respect to 10 then 9 then 8 and so on and

%so fourth in one second segments

Hw 7 problem 4:

I only changed one part in the PlayTurn section of the tic tac toe game in order for a computer to randomly select numbers instead of there being a player. This part took into consideration if it was X or O’s turn and if it was O’s turn then it would make the move, I did this by creating an if else statement, on the part where the player is supposed to move and this statement determined if it was the players turn or not.

% Nic

% 3-4-2014

% Math 50

% Lecturer: Derek

%

% This script will run a one-player, tic-tac-toe game.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialization

clear all

B = zeros(3,3); %the game board

GameOver = false;

for turn = 1:9

clc

ShowBoard(B);

B = PlayTurn(B,turn);

GameOver = CheckBoard(B);

if GameOver

break

end

end

DisplayWinner(B,turn);

function [ ] = ShowBoard( B )

% INPUT: game board B

% OUTPUS: [none: a void function]

A = repmat(' ',[3 3]);

% 'X' and '0' translation

for j = 1:3

for k = 1:3

if B(j,k) == 1

A(j,k) = 'X';

elseif B(j,k) == -1

A(j,k) = 'O';

else

A(j,k) = ' ';

end

end

end

disp('Current board:')

disp(sprintf('%s|%s|%s\n',A(1,1),A(1,2),A(1,3)));

disp(sprintf('-+-+-\n'));

disp(sprintf('%s|%s|%s\n',A(2,1),A(2,2),A(2,3)));

disp(sprintf('-+-+-\n'));

disp(sprintf('%s|%s|%s\n',A(3,1),A(3,2),A(3,3)));

disp('Locations:')

disp(sprintf('1|2|3\n'));

disp(sprintf('-+-+-\n'));

disp(sprintf('4|5|6\n'));

disp(sprintf('-+-+-\n'));

disp(sprintf('7|8|9\n'));

end

I changed something in here:

function [ B ] = PlayTurn( B, turn )

% INPUTS: game board B, turn number

% OUTPUT: revised board B

if mod(turn,2) == 1

disp('It is X''s turn.')

else

disp('It is O''s turn.')

end

tolerance = 5; %to prevent an infinite loop

while tolerance > 0

if (mod(turn,2)~=1)

A = randperm(9);

pick = A(1);

%I only added this if statement to make the computer move the

%circles because circles are represented when mod(turn,2)~=1 also

%because i make A a random assortment of the numbers between 1 and

%9 and the code forces a player to repeat their selection if it

%isn't applicable, the computer will always move.

else pick = input('Where do you want to make your move? ');

end

if isnumeric(pick)

pick = round(pick);

if pick >=0 && pick <= 9

row = ceil(pick/3);

col = mod(pick,3);

if col == 0

col = 3;

end

if B(row,col) ~= 0

disp('That spot has already been used')

else

B(row,col) = (-1)^(mod(turn,2) + 1); %play 'X' or 'O'

break

end

end

end

tolerance = tolerance - 1;

end

end

function [ ] = DisplayWinner( B, turn )

% INPUT: game board B, turn number

% OUTPUT: [none: void function]

clc

ShowBoard(B);

if CheckBoard(B)

if mod(turn,2) == 0

disp('2P is the winner!')

else

disp('1P is the winner!')

end

else

disp('Cat''s Game (no winner)')

end

end