E5 Lab 7

Name:__________Answer__________________

Due in lab on Thursday 10/29/2009

Turn in a hardcopy of this document, not an electronic copy, when you come to lab. The directions for the lab are at:

http://www.swarthmore.edu/NatSci/echeeve1/Class/e5/E5Lab7/E5Lab7.html

Fill in this document. Where appropriate, cut and paste your results from MatLab displayed in bold and in italics. Save often (to your user folder) so you don’t lose your work.

1) In the space below paste:
   a) your well-commented code for solveTPuzzle.m, and
   b) the image.

Note: I used a small font so that the comments would not wrap around to a new line.

clear all
load shapes
numPieces=size(shapes,3);
theta=[0 90 -90 45];
tx=[1 1 -1 0];
ty=[3 0 4 2];
cla;
axis(5*[1 1 -1 1],'square'); grid on;

theta values for solutions

numPieces values

For i=1:numPieces,
[x,y,c] = pickShape(shapes, i);
p(i) = patch(x,y,c,'EraseMode',emode);
end

for i=1:numPieces,
[xnew, ynew] = rotxy(x,y,theta(i),tx(i),ty(i));
set(p(i),'Xdata',xnew,'Ydata',ynew);
end

Figure 1
2) Paste your well-commented code for Task1.m. You will be graded on the quality of your commenting.

```matlab
clear all
load shapes
numPieces=size(shapes,3);
theta=[0 90 -90 45];
tx=[1 1 -1 0];
ty=[3 0 4 2];
cla;
axis(5*[-1 1 -1 1], 'square'); grid on;
title('Task 1');

emode='xor';
% clear all variables
%load the shapes matrix (from lab 6)
%determine how many pieces we have
%Rotation for each of the four pieces
%Translation in x for the four pieces
%Translation in y
%The next section of code creates an array of patches.
%   p(1) is the first patch...
%   This way we can access the patches through their indices.
for i=1:numPieces,
	[x,y,c] = pickShape(shapes, i);
	p(i) = patch(x,y,c, 'EraseMode',emode);
end
numSteps=20; %The number of steps in the animation.
for i=1:numPieces,
	thetaA=linspace(0,theta(i),numSteps);
	xA=linspace(0,tx(i),numSteps);

tyA=linspace(0,ty(i),numSteps);
	[x,y,c] = pickShape(shapes, i);
	for j=1:numSteps,
		[xnew, ynew] = rotTrans(x,y,thetaA(j),txA(j),tyA(j));
	set(p(i), 'Xdata',xnew, 'Ydata',ynew);
	pause(0.1);
end
end
```

3) Which is your favorite erase mode, and why? (This is just here to make sure you experimented with the different modes)

I like 'xor' because there is no loss of information. You can see where all pieces are at all times (they are never obscured behind another shape).

4) Paste your well-commented code for Task2.m. You will be graded on the quality of your commenting.

```matlab
clear all
load shapes
numPieces=size(shapes,3);
theta=[0 90 -90 45];
tx=[1 1 -1 0];
ty=[3 0 4 2];
cla;
axis(5*[-1 1 -1 1], 'square'); grid on;
title('Task 2');

emode='xor';
% The next section of code creates an array of patches.
```
% p(i) is the first patch...
% This way we can access the patches through their indices.
for i=1:numPieces,
    [x,y,c] = pickShape(shapes, i);
    p(i) = patch(x,y,c,'EraseMode',emode);
end

numSteps=20; %The number of steps in the animation.
for i=1:numPieces, %Iterate through pieces
    if theta(i)<0, %If theta(i)<0...
        theta(i)=theta(i)+360; % add 360 to it to make it >0.
    end
    thetaA=linspace(0,theta(i),numSteps); %Generate intermediate angles
    txA=linspace(0,tx(i),numSteps); %Intermediate tx...
    tyA=linspace(0,ty(i),numSteps); %Intermediate ty.
    [x,y,c] = pickShape(shapes, i); %Get x and y values for ith piece
    for j=1:numSteps, %Find rotated and translated x and y for the intermediate values
        theta(j)=thetaA(j);
        txA(j)=txA(j);
        tyA(j)=tyA(j);
        [xnew, ynew] = rotTrans(x,y,theta(j),txA(j),tyA(j));
        set(p(j),'Xdata',xnew,'Ydata',ynew'); %Move the patch to the new coordinates
        pause(0.05); %Pause to show new positions
    end
end

5) Paste your well-commented code for Task3.m. You will be graded on the quality of your commenting.

clear all %clear all variables
load shapes %load the shapes matrix (from lab 6)
numPieces=size(shapes,3); %determine how many pieces we have
theta=[0 90 -90 45]; %Rotation for each of the four pieces
tx=[1 1 -1 0]; %Translation in x for the four pieces
ty=[3 0 4 2]; %Translation in y
cla; %clear the current figure
axis([5*[-1 1,-1 1],'square']); grid on;
title('Task 3');
emode='xor'; % try 'xor', 'none', 'background', 'normal'

%The next section of code creates an array of patches.
% p(i) is the first patch...
% This way we can access the patches through their indices.
for i=1:numPieces,
    [x,y,c] = pickShape(shapes, i);
    p(i) = patch(x,y,c,'EraseMode',emode);
end

numSteps=20; %The number of steps in the animation.
for i=1:numPieces, %Iterate through pieces
    thetaA=linspace(0,theta(i),numSteps); %Generate intermediate angles
    txA=linspace(0,tx(i),numSteps); %Intermediate tx...
    tyA=linspace(0,ty(i),numSteps); %Intermediate ty.
    [x,y,c] = pickShape(shapes, i); %Get x and y values for ith piece
    for j=1:numSteps,
        %Rotate piece, keeping translation in x and y equal to zero..
        [xnew, ynew] = rotTrans(x,y,thetaA(j),txA(j),0,0);
        set(p(i),'Xdata',xnew,'Ydata',ynew'); %Move the patch to the new coordinates
        pause(0.03); %Pause to show new positions
    end
for j=1:numSteps,
    %Translate in x, keeping last thetaA value (I could also use
    % "theta(i)" instead of "thetaA(end)"). keeping ty=0.
    [xnew, ynew] = rotTrans(x,y,thetaA(end),txA(j),0,0);
    set(p(i),'Xdata',xnew,'Ydata',ynew'); %Move the patch to the new coordinates

pause(0.03);  %Pause to show new positions
end
for j=1:numSteps,
%Translate in y, keeping last thetaA and txA value
[xnew, ynew] = rotTrans(x,y,thetaA(end),txA(end),tyA(j));
set(p(i),'Xdata',xnew,'Ydata',ynew');  %Move the patch to the new coordinates
pause(0.03);  %Pause to show new positions
end

6) Optional – extra – only do it if you want a bit of a challenge. Paste your well-commented code for Task4.m. You will be graded on the quality of your commenting.

7) In the space below paste:
   a) the values for the slope and intercept (ma and ba)
   b) your well-commented code for Task5.m (including code to show best fit line and legend), and
   c) the finished plot with your name appended to the title of the plot.

   ma = 0.0856
   ba = -127.4000

   pw=[1500 1000 1250 1750 2000];  %pulse width in microseconds
   thetaA=[-4 -41 -18 22 46];  %angle (degrees)

   plot(pw,thetaA,'bs');
xlabel('Pulse width (\muSeconds)');
ylabel('Angle \theta_a (degrees - \celsius)');
title('Pulse width vs. angle of motor A - Erik Cheever');
axis([500 2500 -60 60]); grid on;

   p=polyfit(pw,thetaA,1);  %Do a first order (linear) fit of the data
   ma=p(1)  %slope
   ba=p(2)  %intercept

   myPW=[1000 2000];  %Pulse width values - two points define the line.
   myTheta=ma*myPW+ba;  %Corresponding "theta" values.

   hold on;  %Don't Erase before next plot
   plot(myPW, myTheta, 'b:', 'LineWidth',2);  %Plot points. Note: I also made %line thicker (not required)
   legend('Data','Best Fit');
   hold off;  %Do Erase before next plot
8) In the space below paste:
   a) the MatLab output when you enter the command “[pa bp] = getPWs(-40, 0)” at the MatLab prompt and
   b) your well-commented code for `GetPWs.m`.

   ```matlab
   >> [pa pb]=getPWs(0,-40)
   pa =
       1.4883e+003
   pb =
       1.1068e+003
   function [pwa pwb] =  getPWs(thetaA, thetaB);
   % The input to this function are two angles.
   % These angle conform to lines given by:
   %    thetaA = ma*pwa + ba
   %    thetaB = mb*pwb + bb
   % The function returns the proper values of pwa and pwb.
   % The values of ma, ba, mb, and bb are given below.
   ma=0.0856;  ba=-127.4;
   mb=0.0833;  bb=-132.2;
   pwa = (thetaA-ba)/ma;  %find pwa given thetaA
   pwb = (thetaB-bb)/mb;  %find pwb given thetaB
   ```