CS1112-206/211/212–Fall 2009
Section 11 Solution
Friday, November 13

1 Triangle Perimeter

To compute the perimeter of T, we need the lengths of all the three sides. We can obtain these by using function `Dist`, giving it the two endpoints (which is encoded in T) of each side. We then simply add the results up.

```
d=Dist(T.A,T.B)+Dist(T.B,T.C)+Dist(T.C,T.A);
```

2 Middle Triangle

We simply need to calculate the midpoint of each side, using function `Mid`, and create a new triangle structure with these points. We obtain the color of the new triangle from T.

```
S=MakeTriangle(Mid(T.A,T.B),Mid(T.B,T.C),Mid(T.C,T.A),T.color);
```

3 Shaded Triangle

To draw a closed figure with color, the `fill` function is the way to go. We need to make sure to pass in the parameters correctly: The first parameter should be the x-coordinates of the vertices, the second parameter should be the y-coordinates *in the same order as the first paramater*, and the third parameter is the color, which we obtain from T. The key statement is shown below:

```
fill([T.A.x T.B.x T.C.x],[T.A.y T.B.y T.C.y],T.color);
```

4 Nearly Right

This problem is almost the same as one we did in Lab 2, though we allow some errors here. First, obtain the lengths of the three sides using function `Dist`. Then, pick two of the sides, add their squares, and compare the sum to the square of the remaining side. If the difference is at most 10^{-6}, assign `alfa=1;`. Otherwise, assign `alfa=0;`. Alternatively, we can determine which side is longest first, and then add the squares of the two remaining side lengths to compare with the square of this side length.

5 Shifted Triangle

To shift a triangle horizontally, we need to modify the x-coordinates of the three vertices. We can do this by, first, duplicate the structure by assigning T to S. Then, the key idea here is that any

change to S will not affect T. Now, we can simply adjust the x-coordinate of each point in S by accessing `S.A.x` for point A, for example.

6 Random List of Triangles

We need to randomize n triangles, so a `for` loop is needed here. For each triangle, we have to sample three points (to be the vertices) from the area $(0,1) \times (0,1)$. This can be done using `rand` function twice, one for each coordinate. Once we have all the three points, we can simply create a structure and append it to the structure array we have created so far.

Note that it is possible that two of the three points are identical, or that all the three points lie on the same line. In that case, what we have randomized will not form a triangle. This happens with an extremely low probability, so we do not have to worry about that. Nonetheless, we can prevent this by checking whether any of these are true. In that case, the code in the `for` loop will be as follows:

```
A=MakePoint(rand(1),rand(1));
B=MakePoint(rand(1),rand(1));
C=MakePoint(rand(1),rand(1));

while ~isTriangle(A,B,C)
    A=MakePoint(rand(1),rand(1));
    B=MakePoint(rand(1),rand(1));
    C=MakePoint(rand(1),rand(1));
end

T=[T MakeTriangle(A,B,C,'r')];
```

where the function `IsTriangle` is as follows:

```
function alfa=isTriangle(A,B,C)
% Determine whether three points form a triangle.
% A, B, and C are points in the xy-plane
% alfa is 1 if these points form a triangle and 0 otherwise.

% check if any two points are the same
if (A.x==B.x && A.y==B.y) || (A.x==C.x && A.y==C.y) || (B.x==C.x && B.y==C.y)
    alfa=0;
else
    % check if these three points are on the same line
    % Let A be a reference point.  If the slope of the line connecting A
    % and B is the same as the slope of the line connecting A and C, then
    % the three points are on the same line.

    % calculate slope of AB
    if A.x==B.x
        mAB=inf;
```

```
    else
        mAB=(A.y-B.y)/(A.x-B.x);
    end
    % calculate slopt of AC
    if A.x==C.x
        mAC=inf;
    else
        mAC=(A.y-C.y)/(A.x-C.x);
    end

    if mAB==mAC
        alfa=0;
    else
        alfa=1;
    end
end
```