

CS1115 Fall 2013 Project 6 Due Tuesday November 26 at 11pm

You must work either on your own or with one partner. You may discuss background issues and general solution strategies with others, but the project you submit must be the work of just you (and your partner). If you work with a partner, you and your partner must first register as a group in CMS and then submit your work as a group. Each problem is worth 10 points. One point may be deducted for poor style. No partners are allowed for the Challenge problem.

Objectives

This assignment is about recursion, sound processing, image processing, and more advanced GUI design.

1 Volume Estimation (10 Points)

This problem is about a 2-dimensional version of the recursive trapezoidal rule procedure illustrated in `ShowDefIntegral` that is available from the syllabus page. The goal is to integrate a function $f(x, y)$ over a triangle. We will assume that f is positive so we can think of the calculation as that of computing the volume of a triangular building whose roof has height $f(x, y)$ over floor coordinate (x, y) .

Just as the trapezoidal computation

$$\text{Area} = (\text{length of line segment base}) \times (\text{average height}) = (R - L) \frac{f(L) + f(R)}{2}$$

is the basis for the trapezoidal rule, the prismoidal¹ computation

$$\text{Volume} = (\text{area of triangular base}) \times (\text{average height}) = A \cdot \frac{f(x_1, y_1) + f(x_2, y_2) + f(x_3, y_3)}{3}$$

is the basis of the prismoidal rule. Here, A is the area of the triangle with vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) . You are to implement a recursive function

```
function V = Volume(u,v,f,fVals,tol,Level)
% u and v are row 3-vectors and (u(1),v(1)), (u(2),v(2)), and
% (u(3),v(3)) are the vertices of a triangle T.
% f is a function of two variable and f(x,y) is defined on T.
% fvals is a row 3-vector where fvals(i) = f(x(i),y(i)), i=1:3
% tol is a positive real number
% V is an approximation of the integral of f over T.
% Level is the number of allowable levels of subdivision.
```

that is analogous to the recursive trapezoidal method. Here is the main idea. Let V be the prismoidal volume over triangle T . Now by connecting each vertex of T to its centroid, we obtain three smaller triangles T_1 , T_2 , and T_3 . Let V_{better} be the sum of the prismoidal volumes associated with these three triangles. If `Level==0` or if

$$|V - V_{better}| \leq \text{tol} \cdot \text{area of } T,$$

then `Volume` should return V_{better} . Otherwise, it should call itself three times to compute the prismoidal volumes associated with T_1 , T_2 , and T_3 and it should return the sum of these approximations. Submit `Volume` to CMS.

2 Phone GUI (10pts)

Implement a GUI that can be used to simulate the interaction with a "pay" phone. It should have the following features.

¹I made this term up!

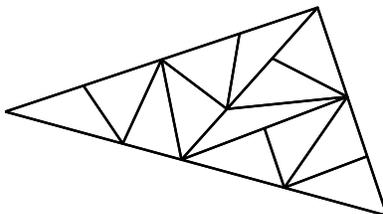
1. It should have a 4-by-3 array of push buttons as you see on page 327 in FVL. (The little letters do not have to be displayed on the buttons.)
2. When a button is depressed, the appropriate tone should be realistically played. Read section 13.2 in FVL. You are free to use or modify any of the FVL codes.
3. There should be an edit box that functions as an LED display so that the telephone number is displayed as it is being entered via the button pushes.
4. The number-entering process should terminate if the user enters '911' or after 7 digits have been entered provided the first button push is not a "1" or after 10 digits have been entered provided the first button push is a "1" or if the "hang-up" button is depressed. (The hang-up button is discussed below.)
5. A call can be made after the entering process completes AND a coin has been inserted. Use a checkbox to simulate the latter. Two things should happen when a call is made. First, the button-push tones should be realistically played back in order. Second, a sound bite should be played. Visit <http://www.mediacollege.com/downloads/sound-effects/> which has a huge library of free .wav sound effects. You can play back anything you like as long as it is no longer than ten seconds. The sound of a ringing phone is an obvious choice.
6. Every phone call should be added to a "list of contacts". Use a pop-up menu for this. If a number is selected from the pop-up menu and the coin checkbox indicates payment, then the call should be made. If a number is selected from the pop-up menu and the checkbox indicates that no payment has been made, then play some other (insulting!) sound effect. (Download `ShowPopUp.zip` which contains an instructive GUI.)
7. If a local call is made, then a .jpg image of your choice should be displayed in an appropriately-sized axes component. This component should only be visible while the call is being made.
8. There should be a "hang-up" button. When it is depressed the "dialing process" should be reset, i.e., the "LED display" should be cleared and the coin checkbox should be set to its no-pay state with a short sound effect.

Don't try to implement everything at once. Get one feature working before you move on to the next. Zip all your files into `FinalGUI.zip` and submit to CMS.

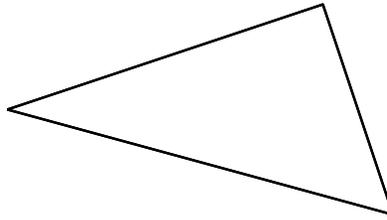
Challenge 6. Equal-Area Partitioning

In this problem you are given a triangle T and are asked to subdivide it into N smaller triangles, each with the same area, e.g.,

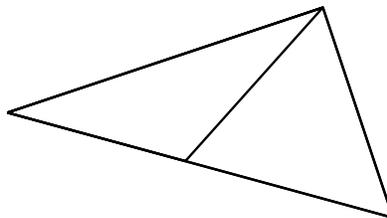
$N = 13$



We outline a way to do this recursively. Call the original triangle S :



Let A be its area. Now draw a line segment from the largest angle to the opposite side in such a way that one triangle (call it S_1) has area $(6/13)A$ and the other triangle (call it S_2) has area $(7/13)A$:



Call this a $(6/13)$ -to- $(7/13)$ split. We then repeat the process on both S_1 and S_2 . We need a $(3/6)$ -to- $(3/6)$ split of S_1 and a $(3/7)$ -to- $(4/7)$ split of S_2 (Why?) By carving up the largest angle we get a nicer overall partition.

You are to write a recursive function that carries out this process:

```
function T = Partition(S,N)
% S is a triangle and N is a positive integer.
% T is a length-N structure array of triangles with the property that
% each T(k) has the same area and that T(1),...,T(N) collectively ‘tile’ S.
```

It must make effective use of these structures:

```
function P = MakePoint('x',x,'y',y)
% x and y are real numbers that are the coordinates of a point P in the plane
P = struct('x',x,'y',y);

function T = MakeTriangle(P1,P2,P3)
% P1, P2, and P3 are points that are the vertices of triangle T
T = struct('P1',P1,'P2',P2,'P3',P3);
```

Submit `Partition` to CMS. It is OK for it to have subfunctions (but not `MakePoint` and `MakeTriangle`).

Note. Given a real number f that satisfies $0 < f < 1$, there is an easy way to calculate the partitioning line segment in order to realize an f -to- $(1 - f)$ split. Figure this out by drawing a picture and recalling that the area of a triangle is one-half base times height.