

08.72.64 Tölvugrafik

Final exam

Teacher: Hjálmtýr Hafsteinsson

May 3rd, 2006

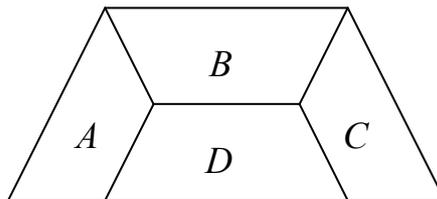
Time: 9⁰⁰-12⁰⁰

All the problems have the same value. You only have to solve 5 problems out of 6. The best five solutions count.

All written material and a calculator are allowed.

1. In this problem it is sufficient to work with 2D transformations.
 - a) Are the following pairs of transformations commutative, i.e. for the transformations T_1 and T_2 , is $T_1 T_2 = T_2 T_1$? Justify each instance.
 - i) Two translations
 - ii) Rotation and translation
 - iii) Two scalings
 - b) Reflection about a coordinate axis can be done by scaling the corresponding coordinate by -1 . Show you can reflect about a line that is given by the 2D points P_1 and P_2 .

2. In the same way that the Sierpinski gasket can be drawn with repeated triangles (see section 2.9 on pages 77-80 in the textbook) isosceles trapezoids, as shown in the figure below, can be used to draw a fractal image.



The figure shows one large isosceles trapezoid consisting of four smaller ones, A , B , C , and D . All four smaller trapezoids are of the same size (the drawing might not be perfectly accurate!) and the ratio between the length of their parallel sides is 1 against $\sqrt{2}$ ($\sim 1.4142\dots$). The same ratio is between the length of the parallel sides in the large trapezoid (i.e. the length of the longer (bottom) side is $\sqrt{2}$ -times more than the length of the shorter side).

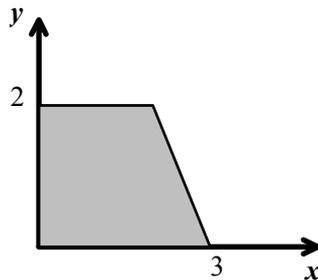
Show the main functions in a OpenGL program that constructs an image, similar to the Sierpinski gasket by recursively filling in the trapezoids A and C . Also explain with a drawing and in words how your program works.

3. When light is reflected off water at night, then the reflection is long and narrow as can be seen in the picture below.

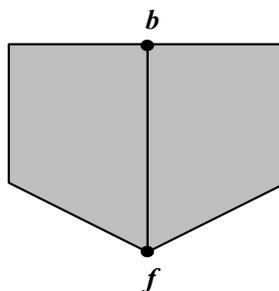


- Explain the shape of the reflection from what you know about perspective transformations, the Phong reflection model, and the properties of water.
- If the water surface was perfectly smooth what would the reflection look like and why?
- Name some other surfaces with similar reflection properties.

4. You are given the function `Wing()` that draws the wing shown in the figure below. It lies in the $z=0$ plane.



- Write the OpenGL function `Butterfly(fx, fy, fz, bx, by, bz)`, that uses `Wing()` to draw the butterfly below. The function receives as parameters the coordinates of the points f , which is the position of the front-end of the butterfly, and b , which is the position of its back-end. Note that the size of the butterfly is determined by the distance between b and f . Use `glPushMatrix()` and `glPopMatrix()` in the right places so as not to influence other parts of the program.



- Explain how the function in part a) can be changed so that the wings can rotate against each other up to 90° about the axis that f and b form, to simulate the flapping of butterfly wings.

5. If the *billboarding* method is used to draw a tree, then a tree texture is mapped to a square, with the alpha values set to 1.0 where the tree is in the texture, and 0.0 everywhere else. With the help of blending, objects behind the “tree” can be seen through it in the places where alpha is 0.0. This method works if the tree is drawn after all other objects. However, if an opaque object behind the tree is drawn after the tree is drawn, then it will not display correctly.

- a) Explain what happens in this instance and what the problem is.
- b) In OpenGL an *Alpha test*, can be enabled. It allows the user to selectively display individual pixels based on their alpha value. The function `glAlphaFunc` is used to define conditions on the alpha values of pixels that they have to fulfill in order to be displayed. For example, `glAlphaFunc(GL_GREATER, 0.5)` only displays those pixels of a polygon with alpha values greater than 0.5. Explain how the alpha test could be used to implement billboarding where the drawing order of the polygons does not matter (i.e. fix the problem described above).

6. In an OpenGL program the viewer should fly around the model following a cubic Bezier curve and it should always face forward along the curve. The viewer will start its trip in the control point P_0 and end it in P_3 .

If you are give the four control points, explain exactly how you would position the viewer and decide its orientation. Use drawings, calculations and OpenGL commands.