

EXAM IN
COMPUTER GRAPHICS
TSBK07

Time: 2th of June, 2010, 14-18

Room: U1/T2

Teacher: Ingemar Ragnemalm,
visits around 15 and 17

Allowed help: None

Requirement to pass: Grade 3: 21 points
Grade 4: 31 points
Grade 5: 41 points

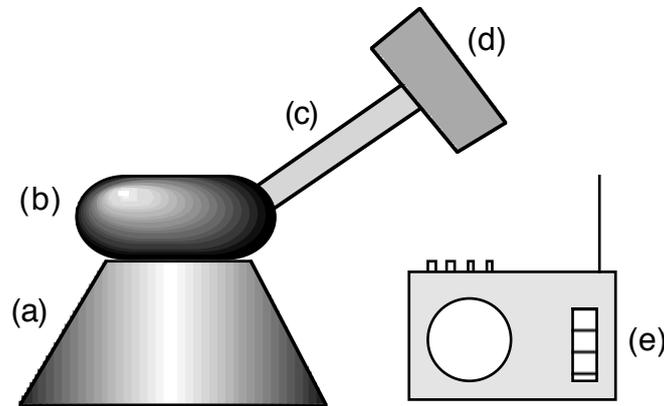
ECTS:
C: 21 points
B: 31 points
A: 41 points

Answers may be given in swedish or english.

Good luck!

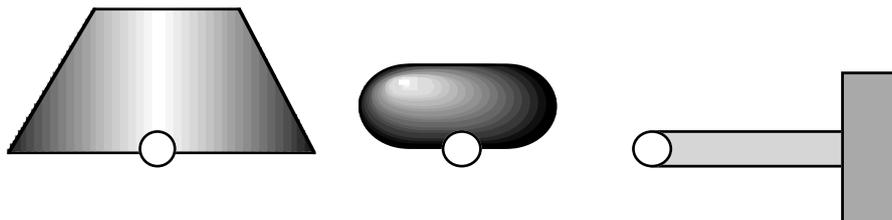
1. OpenGL programming

a) The figure shows a *radio silencer*. It has a two-part body, a main body (a), and an upper part (b) which can rotate around a vertical axis centered on the object. Attached to the upper body (b) is a shaft (c) with a hammer (d). The shaft can rotate around its fastening point, in order to hit the radio (e), which will after a sufficient number of hits be silent forever.



The hammer shaft is fastened in position $(0, y_h, z_h)$ relative to the origin of the upper body, positioned according to the figure, allowing rotation around the X axis by the angle ϕ . The body should be positioned at (x_b, y_b, z_b) , rotated around the Y axis by α . The top is at height h and rotated relative to the body by β .

The parts, with their respective origins (white circles), are shown in default positions below (Y axis up, Z left).



Function calls `drawBody()` (body), `drawTop()` (top) and `drawHammer()` (hammer with shaft) are provided. Write OpenGL code to draw the silencer in the specified location, with the hammer and top rotated as desired.

(3p)

b) Write OpenGL code to place the camera in $(2, 0, 0)$ looking towards origin, using basic transformations only (*not* `gluLookAt()` or similar utility calls). The code should also including any other code necessary for this specific task.

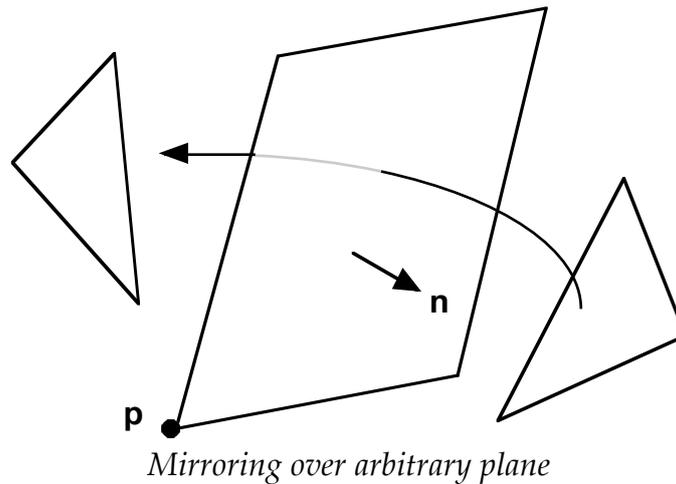
(3p)

c) Describe (in one sentence each) the meaning of *varying*, *uniform* and *attribute* when applied on variables in GLSL.

(2p)

2. Transformations

A plane is given by its normalized normal vector \mathbf{n} and a point \mathbf{p} in the plane. Give a sequence of 4x4 transformation matrices that perform mirroring over this plane. You do not have to multiply the matrices together.



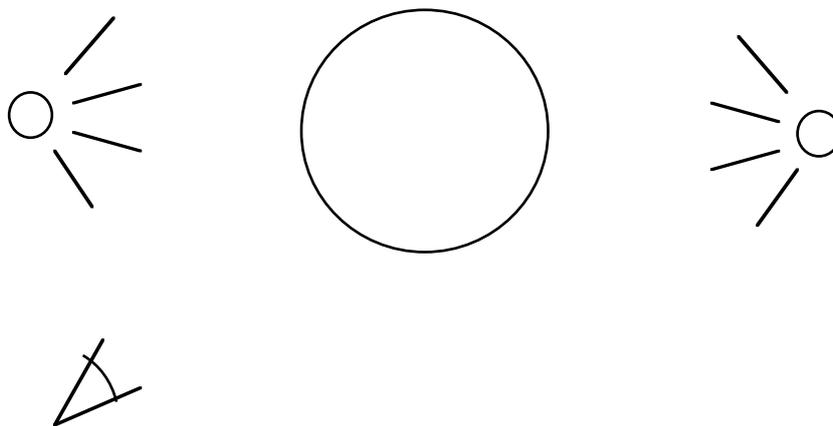
Hint: This is a variation of the "rotation around arbitrary axis" problem and can be solved by similar means.

Hint: Partial score can be awarded for describing the process even without all information in some steps.

(6p)

3. Light, shading and ray-tracing

The figure shows a spherical surface, two light sources and the camera, all located in the same plane. The scene is rendered in a ray-tracer.



The light source positions, camera position and normal vector in any point on the surface are available as well as material parameters.

a) When calculating specular reflections using Phong's model, most vectors involved are trivially calculated by the data given above (by vector subtraction of position vectors). However, you need to calculate one more vector from the others which requires slightly more calculations. Which one? Derive how it can be calculated.

(3p)

b) Write a formula for calculating shading using the three-component lighting model for any point on the surface. Note that the influence of a light source on the far side of the sphere should be correctly handled. Define symbols in a figure. (4p)

c) If another object is introduced in the scene, the two objects should be able to cast shadows on each other. How is that handled in ray-tracing? (2p)

4. Curve generation

a) Describe the Catmull-Rom spline by its most distinctive features. Include continuity of several segments as well as how control points relate to the curve. (3p)

b) In Bresenham's line drawing algorithm, a decision variable p_k is used. Give a geometrical interpretation of its value, that is, where does it come from? (Formulas are not demanded.) (2p)

5. Surface detail

a) When generating texture coordinates for a mesh using cylindrical coordinates, assigning texture coordinates by vertex, a problem occurs. What is the problem and what can be done about it? (2p)

b) Describe how an axial billboard can be implemented. (4p)

6. Miscellaneous

a) Using a figure, illustrate the difference between the odd-even test and the non-zero winding number test for hit testing 2D polygons. (2p)

b) A Julia set can be calculated as follows:

```
for every pixel (x, y) do
  (zr, zi) = scaling of (x, y)
  for i = 0 to maxiterations do
    z = z2 + L
    if |z| > R then Leave
```

What value is used for displaying the fractal? (2p)

7. Collision detection

a) Write pseudocode for how collision tests are performed in the following two cases: (1) Between two AABB's. (2) Between two spheres.

(2p)

b) Outline a method to reduce the number of object-to-object tests in collision detection, in order to reduce even the bounding shape tests.

(2p)

8. Visible surface detection and large worlds

a) BSP trees can be used to implement Painter's algorithm. Describe how the tree is generated.

(3p)

b) The Z-buffer (depth buffer) method conflicts with the use of transparency. One case where it gives undesirable results is in the rendering of a *single* transparent model (let us assume it is convex for simplicity). Why does this happen and what can be done about it?

(2p)

c) When rendering terrains, the number of polygons in view get very large. This can be remedied by *geomipmapping*. Give a summary of how this works. The method has two problems: cracks and popping. How can these problems be handled? Use figures to illustrate your description.

(3p)