EXAM IN

COMPUTER GRAPHICS

TSBK07

(TEN1)

Time:	23st of October, 2014, 8-12
Room:	TER2
Teacher:	Ingemar Ragnemalm, visits around 10
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.

Please make a special note if you followed the course before 2012. Some answers may be slightly different depending on that and I need to know what material you studied (old or new) to make fair scoring.

- Wish us luck!
- I wish you skill!
[Martin Landau, "Mission Impossible"]

1. OpenGL programming

a) Below follows a few lines of GLSL code that your examiner dreamed up one stormy night. Not only is the code incomplete and rather meaningless, but there are some details that will prevent if from working correctly, or even compiling.

```
#version 150
#include ”stdio.h”

in texCoord;
out uniform float[4] gl_fragColor;
uniform float f;

void main()
{
      uniform texture2D myTexture;

      if (f == 0)
            printf(”Warning, f = 0!\n”);
      else
      {
            float gl_s = texCoord[0]/f;
            float gl_t = texCoord[1]/f;
      }
      vec4 color = myTexture[gl_s, gl_t];
      gl_fragColor = color.rgbx;
}
```

What errors or otherwise ”bad” code can you find? A few words explaining the problem for each is enough (like ”divide by zero”). Each error should only be given once. Six errors must be found for full score.

(3p)

b) You have a variable in your host OpenGL program, *GLfloat time*, which holds the time. This value is needed in your vertex shader to control an animation. Describe how the host OpenGL program sends the variable to a shader program. Clarify how variables are identified. A code-like example is preferred.

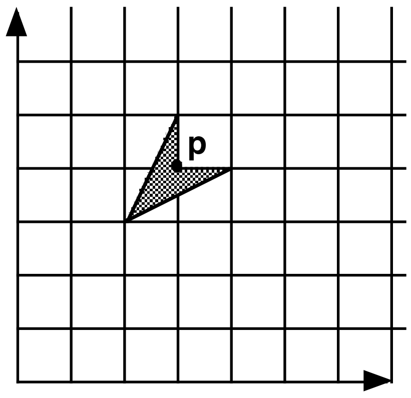
(2p)

c) In order to produce a normal matrix, a transformation matrix for normals, we can take the inverse transpose of... which matrix?

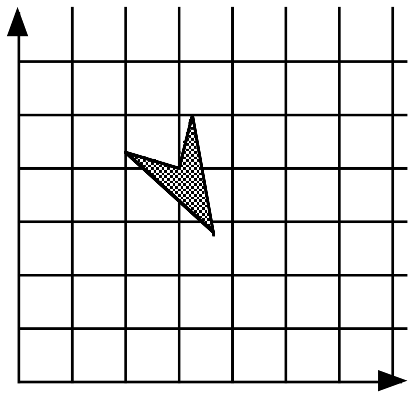
(1p)

2. Transformations

a) In the figure, a 2D shape is shown together with a point **p**. Produce a sequence of 3x3 matrixes that define a transformation that rotates the shape (or anything else) around **p** by an angle ϕ. The contents of each matrix should be given. You don’t have to multiply the matrices together.



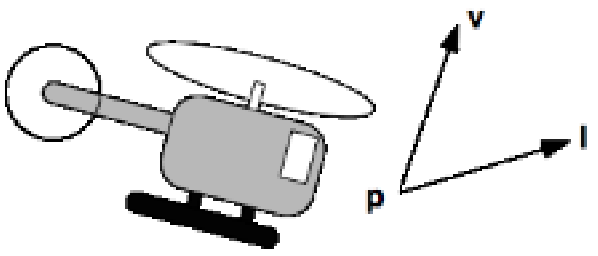
Original shape and position, and the point **p** that the shape is rotated around



After transformation

(3p)

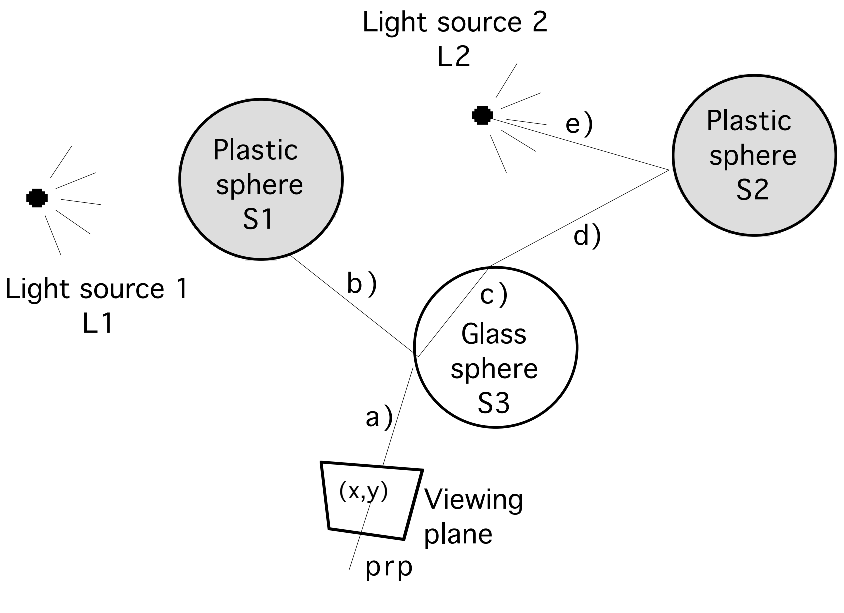
b) You are writing a helicopter game (see figure). You want a first-person view camera that is always aligned with the helicopter. Given the camera position **p**, and a look-at vector **l** pointing to some point on the ground, plus the up-vector of the helicopter **v**, produce a camera matrix that will look in the helicopter's forward direction, but as close to the look-at vector as possible.



Hint: This means that (contrary to the usual case) the up-vector is strict while the look-at vector is approximative.

(4p)

3. Light, shading and ray-tracing



a) Write a formula for the three-component light model, using a figure to clarify symbols. The formula should include handling of surfaces facing away from the light sources as well as multiple light sources.

(3p)

b) A couple of rays (a-e) used to calculate the pixel (x, y) are shown in the figure. Give each ray appropriate descriptive names. How is each ray formed? Are some rays clearly missing? If so, which ones?

(3p)

c) Ray-tracing can be a very time consuming task. Suggest two possibilities to reduce the computation time for a ray tracer where the time can be controlled *by a user defined parameter*. Your two suggested methods should not be variations on the same concept but apply to two significantly different concepts in a ray tracer.

Suggesting one such parameter (relevant to the question) will give partial score.

(3p)

4. Surface detail

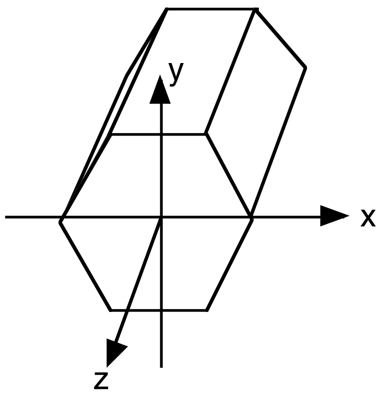
a) Linear coordinates (u, v) can be defined by

$$x = u$$
$$y = v$$

Write formulas for linear texture mapping, mapping x, y, z to texture coordinates (s, t), normalized to the interval [0, 1].

(2p)

b) The formula(s) you wrote in (a) can be used to map a texture (by vertex) onto a shape that is roughly cylindrical, with six sides. The height of the cylinder is in the z direction. The six edges along the z axis are at angles 0, π/3, 2π/3, π, 4π/3 and 5π/3. We don't consider the texture on the end sides.



A six-sided approximation of a cylinder

However, this mapping produces an error. What error? Suggest how we can overcome it.

(2p)

c) A skybox is an easy way to get an illusion of an environment at (for practical purposes) infinite distance. Altough the principle is simple, mapping a texture on a cube which works as a backdrop, there are several important issues that have to be considered in order to make a good skybox. Give *three* important issues. A brief sentence for each should suffice.

(2p)

5. Curve generation

a) Is a Bézier curve an interpolating spline or an approximating spline? Why?

(2p)

b) Two segments of a 2D spline is given by the following functions:

$p_x(u) = -3 + 6u - 2u^2$
$p_y(u) = u + u^2$
$q_x(v) = 1 + 4v + 2v^2$
$q_y(v) = 2 + 6v - 2v^3$

What continuity criteria do these segments fulfill for u=1, v=0?

(3p)

6. Miscellaneous

a) Explain the difference between supersampling and multisampling. A figure is recommended.

(2p)

b) Using a figure, illustrate the difference between the odd-even rule and the non-zero winding number rule. The figure should have at least one area where the two rules give different result.

(2p)

7. Collision detection and animation

a) Given a triangle **a**, **b**, **c** in a plane, how can you calculate the plane's normal vector n?

(1p)

b) Given a line segment given as two points $\mathbf{p_1}$ and $\mathbf{p_2}$, and a plane given by a point **a** in the plane and the normal vector **n**, describe how you can test whether the line segment intersects the plane and if so, where.

(2p)

c) Given a plane and a triangle (given as above) and a point **p** in the in the plane describe how you can test whether that point is in a triangle.

(3p)

8. Visible surface detection and large worlds

a) Describe mathematically how you can perform frustum culling for an object for which an enclosing sphere is given. How many tests are needed?

(3p)

b) Using a figure, describe how the ”cells and portals” VSD method works. What kind of environments is this method most suited for?

(2p)

c) Describe how a view plane oriented billboard can be implemented.

(2p)