EXAM IN

COMPUTER GRAPHICS

TSBK07

(TEN1)

Time:	20th of August, 2015, 8-12
Room:	TER1
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) Placement of models in a 3D scene generally requires several matrices to be multiplied together. Where should this multiplication take place, on the CPU or in a shader? Motivate your answer.

(2p)

b) Your host program can pass data to the vertex shader stage. This is done in two significantly different ways. Which two, and what is the difference between them?

(2p)

c) How do you produce a normal matrix, a transformation matrix for normal vectors, which will work for non-uniform scaling? (Partial score for a transformation that only works with uniform scaling.)

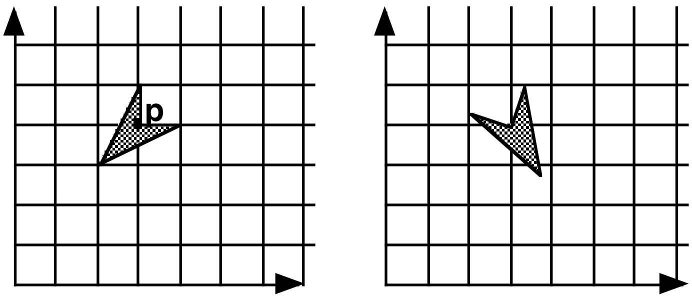
(3p)

2. Transformations

a) In the graphics transformation chain, we have the model-to-world matrix, the world-to-view matrix and the projection matrix. What is each step used for when designing an animation?

(2p)

b) 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.



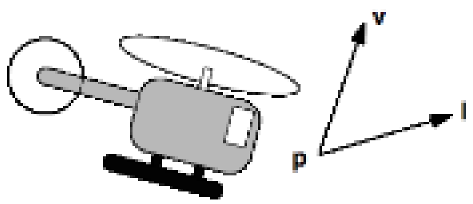
Original shape and position, and the point **p** that the shape is rotated around

After transformation

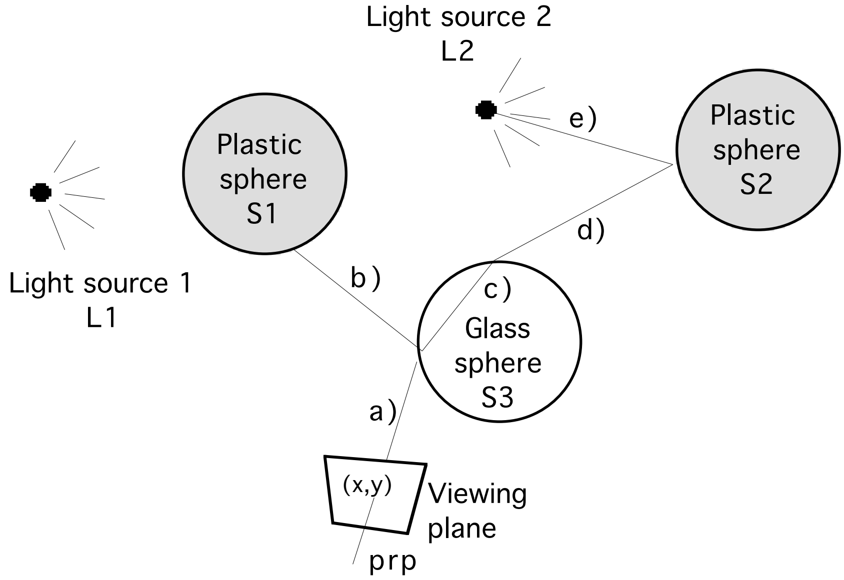
(3p)

c) 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) A student got the task of writing a formula for the three-component light model. The result looked like this:

$$i = i_{amb} + i_{diff} * \cos(N \bullet V) + i_{spec} * \cos(R \bullet L)^n$$

where N is a vector towards the light source, N is the normal vector of the surface, R is L mirrored over N and V is a vector towards the camera. The value i is the resulting intensity, i_{amb} the ambient light level, i_{diff} the light level for the diffuse component and i_{spec} is the light level for the specular component.

I am sure you can improve this, right? Please do! You may either give a better formula or list the errors.

(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) What is the difference, computationally and visually, between Gouraud shading and Phong shading?

(2p)

4. Surface detail

a) Describe how environment mapping works. Include an outline of how you can implement it in OpenGL.

(3p)

b) Describe how mip-mapping works. When accessing a mipmap, what kind of filtering can be performed? Clearly state (or illustrate by a figure) what information this filtering involves.

(2p)

5. Curve generation

a) Is a Catmull-Rom 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) Supersampling and multisampling are two closely related methods for anti-aliasing. Outline how they work. How are they different?
(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)

c) How can you determine, mathematically, whether a geometric fractal is a well-behaved, converging fractal or diverging? Hint: $ns^D = 1$.
(2p)

7. Collision detection and animation

a) Suggest two bounding shapes suitable for broad phase collision tests of *elongated shapes*.
(2p)

b) Describe a way to accelerate collision detection in a scene with a large number of moving objects. How does the computational complexity change?
(2p)

8. Visible surface detection and large worlds

a) Level-of-detail of models can be processed with methods like edge collapsing and vertex removal. What is the difference? Suggest an advantage of each.
(2p)

b) A skybox is an easy way to produce a convincing environent around your scene. Despite the simplicity of the method, there are several problems to consider to make a good skybox. One is simply to have a proper shape with the right texture coordinates... but what else is there to consider? You should be able to give two more problems.
(2p)

c) Describe how a view plane oriented billboard can be implemented.
(2p)