EXAM IN

COMPUTER GRAPHICS

TSBK07

(TEN1)

Time:	5st of June, 2020, 14-18, extended to 19 for extra time for digitizing and uploading
Room:	Distance
Teacher:	Ingemar Ragnemalm, available by E-mail during exam, and by phone at 070-6262628.
Allowed help:	Any. Copying text from other sources is not allowed.

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.

Upload answers to https://twokinds.se/liu/fileupload/
For assistance on the upload system, write to susanne@twokinds.se

In case of problems with uploading, you can also mail answers to exam@computer-graphics.se

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

1. OpenGL programming

a) In modern OpenGL, we spend much effort in creating buffers and passing data to them, keeping references to these buffers and later passing these to functions. Older OpenGL would rather make function calls passing vertices, normals, colors, texture coordinates etc for immediate use, so called “immediate mode”. Compare these two approaches and assess pros and cons of each.

(3p)

b) A student starting out with OpenGL wrote the following GLSL code. You, who know the subject, look at the code. Suggest what errors there are in the code. A good majority of the errors should be found for full score. You should state what the error is, not just point at a line and say that there is an error.

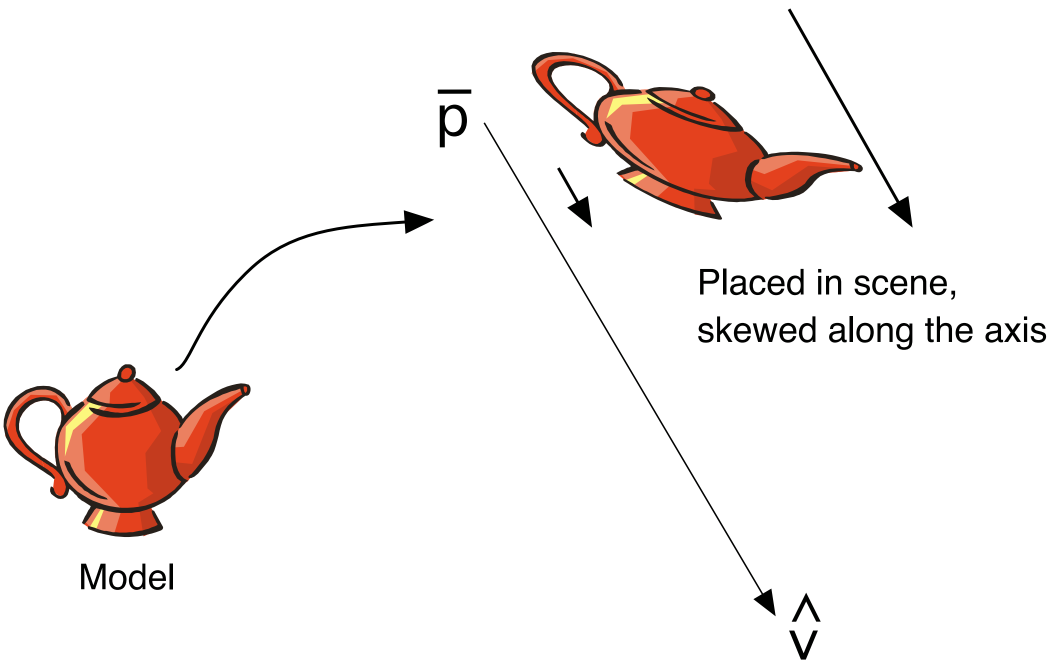
```
#include <stdio.h>

in mat4 m;
in vec3 n;
in vec3 p;
uniform char *a;
out b;
out vec3 n1;

int main(int argc, char *argv[])
{
        printf("%s\n", a);
        vec3 n1 = inverse(transpose(m));
        gl_Position  = p * m;
        b = dot(v, n) + n / n1.xya;
}
```

(3p)

2. Transformations

a) You wish to perform a skewing effect of a part of a scene. The skewing should be performed along a specified line, given as a point p and a vector v. (This implies that the distance to the axis affects how much the position should be displaced along the axis.) See the figure below.

Note: The operation may affect multiple models, possibly the entire scene, not just a single specific model.



You should produce a solution as a sequence of 4x4 matrices. The contents of each matrix should be specified as well as the multiplication order. You do not need to multiply the matrices together.

(4p)

b) Why and when are we using 3x3 matrices or 4x4 matrices, respectively?

(2p)

c) Which of the following statements are true?

1) $T(\mathbf{p}) = T^{-1}(-\mathbf{p})$
2) $R_x(\phi) = R_x^{-1}(-\phi)$
3) $S(s_x, s_y, s_z) = S^{-1}(-s_x, -s_y, -s_z)$
4) $R_x(\phi) * S(2, 1, 1) = S(2, 1, 1) * R_x(\phi)$
5) $T(1, 0, 0) * S(1, 2, 1) = S(1, 2, 1) * T(1, 0, 0)$
6) $R(\boldsymbol{\phi})\, T(\mathbf{p}) = T(\mathbf{p})R(\boldsymbol{\phi})$
7) $S^{-1}(s_x, s_y, s_z) = S^T(s_x, s_y, s_z)$

Note: $T(\mathbf{p})$ = translation matrix. $R_a(\phi)$ = rotation matrix by the angle ϕ around the basis axis $a \in \{x, y, z\}$. $S(s_x, s_y, s_z)$ = scaling matrix. M^{-1} is matrix inverse. M^T is matrix transpose. All matrices are 4x4, as given by the book.

(2p)

3. Light, shading and ray-tracing

a) You are writing a ray-tracer which can be quite time consuming despite you having made optimizations based on scene subdivision as suggested in the course book. Suggest *three* distinctly different features that you can let the user adjust in order to control the workload. How would these affect the quality of the output? Motivate your answer. (Note: Buying faster hardware does not count.)

(3p)

b) What part(s) of the three-component light model is / are irrelevant for a Lambertian surface?

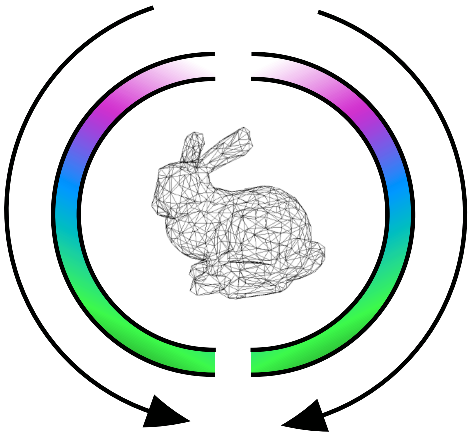
(1p)

c) How are light sources represented in the radiosity model?

(1p)

4. Surface detail

a) You want to make a variant of the mappings that we have seen in the course, a two-sided cylindrical mapping, mapping the entire texture half a turn over the model, and then the *other* way for the back side (see figure below). That is, one side has the texture clockwise and the other counter-clockwise. Describe how such a mapping would work. (Normalized, of course!)



Hint: This can be made as a *simplification* of the usual cylindrical mapping. How, why?

(4p)

b) The following image shows a model that you wish to render in real-time. Suggest what surface mapping techniques that would be relevant for the scene, and for what parts. Motivate your choices.

Special thanks to Susanne for both creating and rendering the model!

(3p)

5. Curve generation

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

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

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

(3p)

b) Compare the DDA line drawing algorithm to Bresenham's line drawing algorithm. What are the merits and potential weaknesses of each? You may argue both from performance and features.

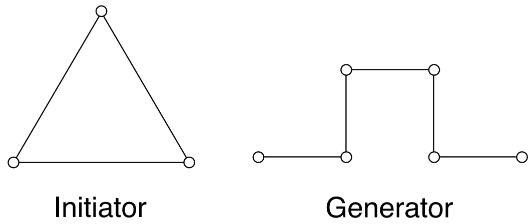
(3p)

6. Miscellaneous

a) Given the assumption that the amplitude falls with 1/f, how much noise suppression can we expect from a 3x3 supersampling? Reasoning with a figure is recommended.

(3p)

b) The following figure describes a geometric fractal. How can we tell whether it is converging or diverging? Use that to determine the behavior of this particular fractal.



Hint: $ns^D = 1$

(3p)

7. Collision detection and animation

a) Sphere-polyhedra is a special case of collision detection where a simplified collision detection applies which is particularly simple, granted that you accept some errors. Describe what kind of errors we get with such a simplified collision detection and argue how severe they are and how you can reduce them.

(3p)

b) Three possible enclosing shapes in the broad phase of collision detection include sphere, AABB and OBB. Compare these three in terms of computational complexity. Which would you use in your own system of these three? Motivate your answer.

(3p)

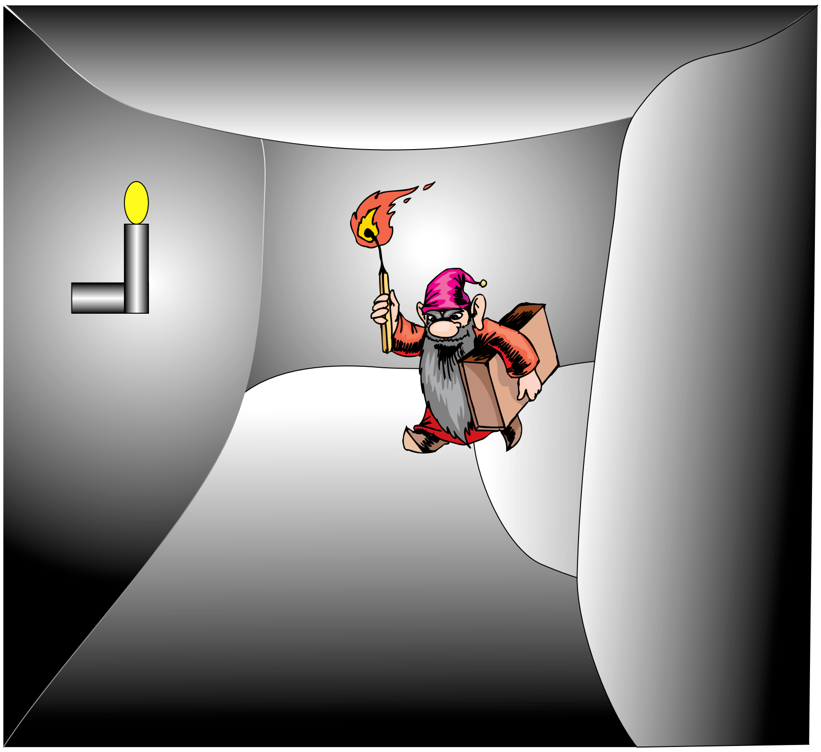
8. Visible surface detection and large worlds

a) A common visible surface detection method has problems with transparency. Which one, why? Suggest how the problem can be avoided in a scene with multiple semi-transparent objects.

(2p)

b) Linus and Linnea are creating an exciting project, a cave exploration game, illustrated by the figure below. The cave involves many twisty little passages, and a large number of wicked-looking dwarves. The cave is lit by torches on the walls, and also by torches carried by the dwarves and the player. All surfaces are texture mapped and lit.

They did no particular optimizations, counting on today's hardware to deal with their game without problems. But the game crawls. Even on the fastest GPUs, it is unplayable.



Suggest *two* remedies that you would say are the most likely to improve rendering speed without significantly ruining rendering quality for this application. Describe the remedies and motivate why they would help.

(4p)