EXAM IN

COMPUTER GRAPHICS

TSBK07

(TEN1)

Time:	18th of August, 2022, 8-12
Room:	TER4
Teacher:	Ingemar Ragnemalm, visits the exam around 10, also available by phone.
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.

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

1. OpenGL and shader 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) In our first OpenGL examples, models were drawn with glDrawArrays(). Motivate why glDrawElements() can be a better choice.

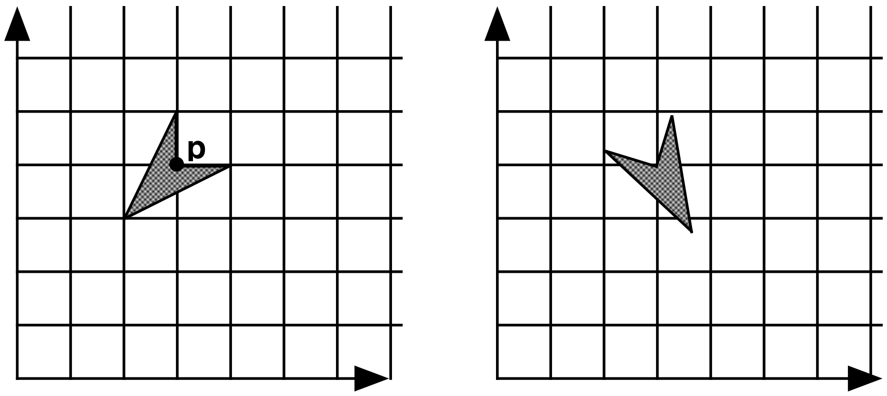
(2p)

2. Transformations

a) Specify how you can construct a camera placement (world-to-view matrix) using only rotations and translations, in order to place the camera in (1, 0, 0) looking at origin.

(3p)

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, using the symbols **p** and ϕ as appropriate. You don’t have to multiply the matrices together.



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

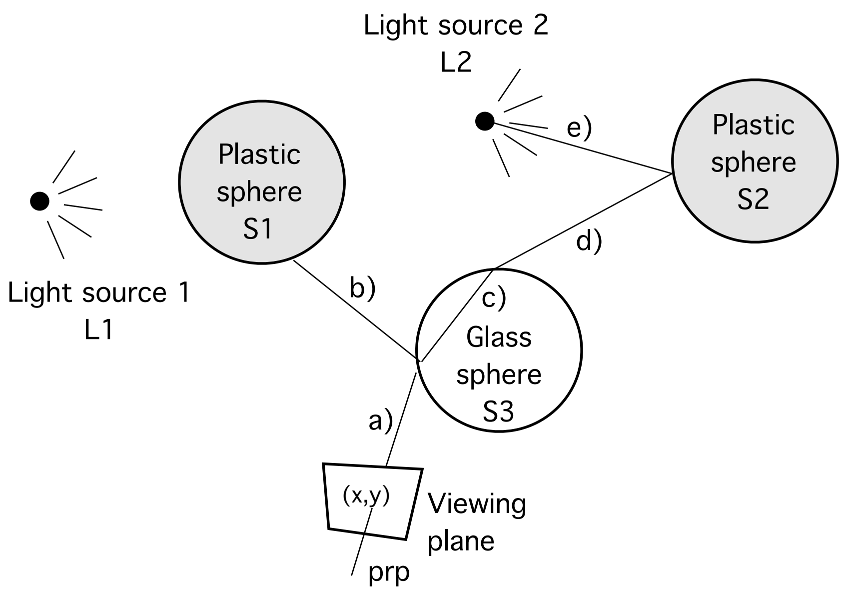
After transformation

(3p)

c) How do you, given a model-to-world matrix, 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.)

(2p)

3. Light, shading and ray-tracing



a) 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)

b) Describe the three-component light model with a figure and a formula, with short text descriptions of each symbol used.

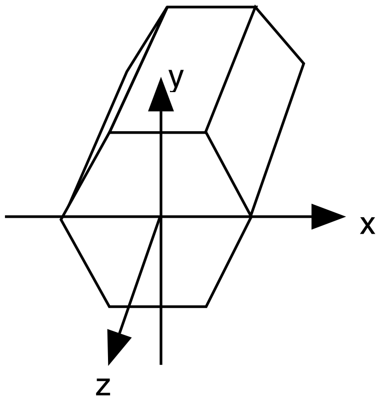
(3p)

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

(1p)

4. Surface detail

a) Cylindrical mapping is 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, $\pi/3$, $2\pi/3$, π, $4\pi/3$ and $5\pi/3$. 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)

b) When implementing a skybox, you need to modify your model-to-view matrix. Why, and how?

(2p)

c) A texture is referenced in two different ways, by a *texture object* and by a *texture unit*. Explain the difference and usage of the two.

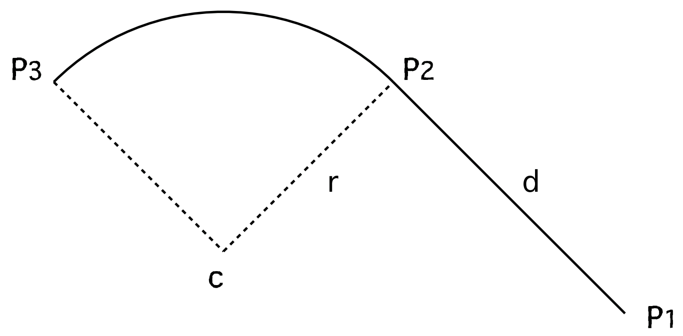
(2p)

5. Curve generation

a) Describe how a cubic Bézier curve can be defined mathematically. Hint: $3u(1-u)^2$

(3p)

b) Two curve segments are described, not by the cubic formulas we are used to, but with a line segment given as the points $\mathbf{p_1}$ to $\mathbf{p_2}$, and a part of a circle centered in **c**, forming an arc from $\mathbf{p_2}$ to $\mathbf{p_3}$. The arc goes from the angle $\pi/4$ to $3\pi/4$.



The line segment from $\mathbf{p_1}$ to $\mathbf{p_2}$ has the length d.
The line segment $\mathbf{c}$ to $\mathbf{p_2}$ is orthogonal to the line segment from $\mathbf{p_1}$ to $\mathbf{p_2}$.

Describe the line segment as a function of u ranging from 0 to 1.
Describe the arc as a function of v ranging from 0 to 1.

Analyze the continuity of the two segments. For full score, a mathematical solution is required. Partial score is given for geometrical observations from the figure.

What value must the radius r be, expressed in the length d, to get C^1 continuity?

Hint: d(sin(a+b*x)/dx = b*cos(a+b*x)

(4p)

6. Miscellaneous

a) The Julia set is defined by the formula

$$z_{k+1} = z_k^2 + L$$

Describe an algorithm that uses this formula to produce a Julia fractal image.

(3p)

b) Describe how multisampling works, using a figure. What is the advantage over supersampling?

(2p)

7. Collision detection and animation

a) Describe how you can handle collision detection and handling between spheres with the same weight.

(2p)

b) Suggest a method to accelerate collision detection in a scene with many moving objects. A figure is recommended.

(2p)

8. Visible surface detection and large worlds

a) A common VSD method has problems with transparency. Describe the problem and a remedy.

(2p)

b) Describe how frustum culling of objects works. The scene contains a large number of objects given as meshes and positions in the world, and uses a world-to-view matrix M.

- The frustum is given by the values *near*, *far*, *left*, *right*, *top* and *bottom*. Convert these values to representations suitable for frustum culling.
- Are there any additional operations that should be done before processing the objects?
- Describe how the tests per object are performed.
- What parts of the frustum should be tested? Is there anything we can skip? Motivate your answer.

(4p)

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

(2p)