EXAM IN

COMPUTER GRAPHICS

TSBK07

(TEN1)

Time:	1st of June, 2013, 8-12
Room:	U1, T2
Teacher:	Ingemar Ragnemalm, visits around 9 and 11
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

in texCoord;
out float[4] gl_fragColor;

void main()
{
	uniform texture2D myTexture;
	uniform f;

	float gl_s = texCoord[0]/f;
	float gl_t = texCoord[1]/f;
	vec4 color = myTexture[gl_s, gl_t];
	gl_fragColor = color.axsw;
}
```

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.

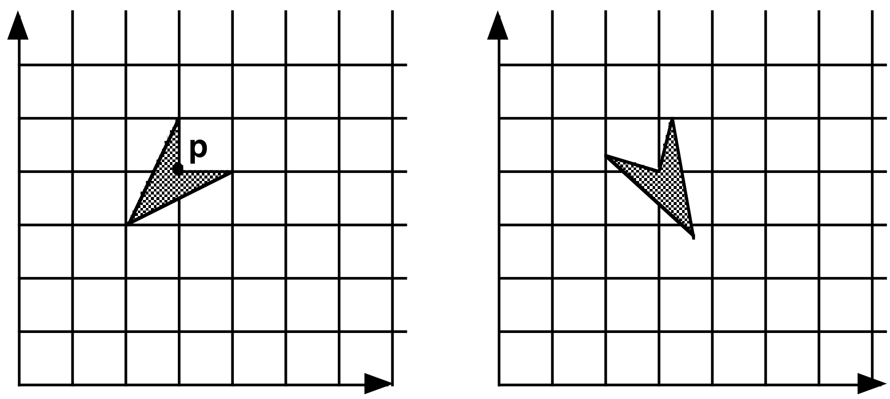
(3p)

b) What is the advantage of using glDrawElements() over glDrawArrays()?

(2p)

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.

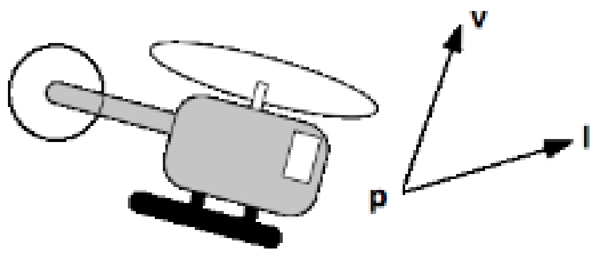


Original shape and position, and the point **p** that the shape is rotated around

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)

c) How do you calculate a normal matrix from a model-to-view matrix?

(2p)

3. Light, shading and ray-tracing

a) 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)

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

(3p)

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

(1p)

4. Miscellaneous

a) Outline a method for generating a random terrain. Any method may be chosen as long as the resulting terrain is reasonably good. Is the method suitable for any of the following?

- Creating a terrain that can be repeated seamlessly to a seemingy infinite world
- After creation, scaling up to a higher resolution using the same method as needed
- Controlling frequency variations in detail

Explain why the method is or is not suitable for each feature.

(4p)

b) Compare supersampling and multisampling in terms of performance and precision. Descibe both sufficiently to clarify the differences.

(3p)

5. Surface detail

a) Texture coordinate generation can be performed by vertex or by fragment. Suggest a distinctive advantage of each (or disadvantage/lack of disadvantage). Motivate your answer.

(2p)

b) When using mip-mapping, several different interpolation modes can be specified. Give an overview of the modes, using a figure to clarify what values are interpolated between.

(3p)

6. Curve generation

a) Two spline segments are given as functions **P**(u) and **Q**(u). They meet in the point **p** = **P**(1) = **Q**(0). Under what conditions does the combined curve fulfill C^1 or G^1?

(2p)

b) Prove that a quadratic Bézier (three-point Bézier) can be expressed as an interpolation of interpolations. The final Bézier can be expressed as:

$$\mathbf{P}(u) = (1 - u)^2\mathbf{P}_0 + 2(1-u)u\mathbf{P}_1 + u^2\mathbf{P}_2,\ u \in [0, 1]$$

Illustrating the process in figures (showing that you understand the principle) can give partial score, while full score requires a mathematical solution.

(4p)

7. Collision detection and animation

a) A line is given by two points $\mathbf{p}_1$ and $\mathbf{p}_2$. A triangle is given by three points **a**, **b**, **c**. Describe how you can test for intersection between the line and the triangle.

(4p)

b) To get good camera placement in an animation, collision detection is vital for the camera. Describe how you can do camera-polyhedra collision detection.

(2p)

8. Visible surface detection and large worlds

a) Describe how a view plane oriented axial billboard can be implemented. The billboard should be located in the point **p** in view coordinates.

(2p)

b) Describe how frustum culling can be applied to a scene with a significant number of spherical shapes (shapes with a known bounding sphere). The camera is placed in the point **prp**, given with scalars for near, far, left, right, top and bottom. A sphere i is located around the point $\mathbf{c}_i$ with the radius r_i, in world coordinates. The world-to-view transformation is given as the matrix M.

What operations need to be done for the test? The method should be reasonably efficient.

(3p)