# EXAM IN

# COMPUTER GRAPHICS

# TSBK07

| | |
|---|---|
| Time: | 1st of June, 2011, 8-12 |
| Room: | T2, U1 |
| Teacher: | Ingemar Ragnemalm,<br>visits around 9 and 11 |
| Allowed help: | None |

Requirement to pass:

Grade 3: 21 points
Grade 4: 31 points
Grade 5: 41 points

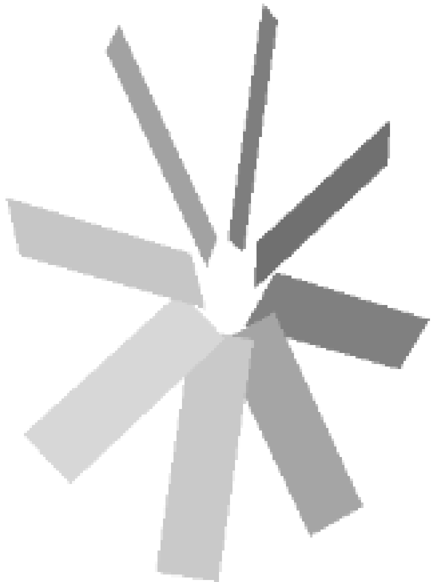
ECTS:
C: 21 points
B: 31 points
A: 41 points

Answers may be given in swedish or english.

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

## 1. OpenGL programming

a) A procedure drawWing() is given (you should not write it). The call draws one wing of the propeller shown below, aligned with the YZ plane (normal vector along the X axis). Using this call, write code to produce an eight-winged propeller, placing the wings around an axis, where each wing should be rotated by itself by 30 degrees (see the figure). (A "for" loop is recommended but not mandatory.)



The propeller center should be placed at ($p_x$, $p_y$, $p_z$) with the axis along the X axis. ~~The camera placement should be restored to its initial state~~. (4p)

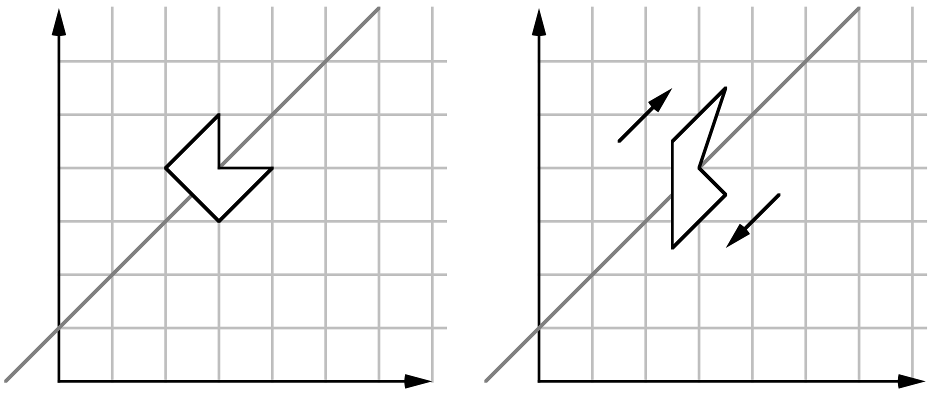
*NOTE: Can be pseudo code, shader not included. Matrix building calls may be used, matrix contents should not be given here.*

b) Phong shading is a classic example for shader programs. Outline how it is performed in a shader program by pseudo code, close-to-GLSL code or a text description. (It does not have to be a complete shader program.) What operations are essential on the normal vector, and why? (3p)

*NOTE: Shader specific things desired.*

## 2. Transformations

a) In the following figures, a shape is skewed along an axis. Give a sequence of 3x3 matrixes, each defining one basic geometric transformation (translations, rotation around origin, scaling and skewing), that define a transformation that produces the result below. Matrix contents should be specified as well as multiplication order. You don't have to multiply the matrices together.

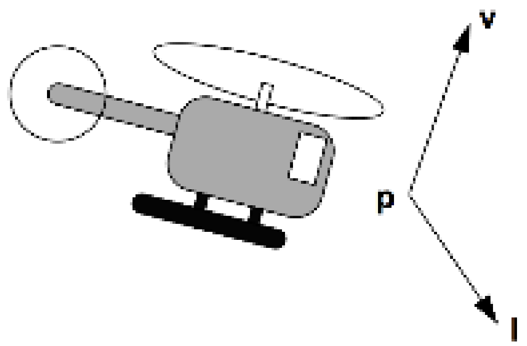


Original shape and position, and the axis the shape is skewed along

After transformation

(4p)

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




Hint: This means that (contrary to the usual case) the given up-vector is exact while the look-at point is approximative. (4p)

## 3. Light, shading and ray-tracing

a) Jittering can produce many interesting effects in ray-tracing. Give descriptive names (or short descriptions) for *three* of them and clarify in a figure how they can be produced. (3p)

b) Write a formula for the three-component light model, using a figure to define vectors. All other variables should be defined by descriptive names. Handling of the far side of objects from the light source should be included. (3p)

## 4. Surface detail

a) Describe, with formulas, how to map a texture to an object vertex by vertex, using a cylinder as intermediate surface. You may use the $\tan^{-1}$ (by one variable). Texture coordinates should be normalized properly. (3p)

b) You are working on a rich scene where much material variations and surface detail is desired. An object in the scene should be a gold nugget, stained with spots of diffuse color (blood or dirt), and a layer of sawdust that has landed on the top. The surrounding scene, a miner's cabin, can be seen as reflections in the gold. The entire object should be rendered as one object (sawdust included). Polygon edges should be as invisible as possible.

What techniques do you need to use to render this object? You do not need to write any code, but you should specify names or brief descriptions of the techniques involved, with respect to what specific effect they have or whether they are of general nature for the whole rendering. (3p)

## 5. Curve generation

a) Under which circumstances will a Bézier curve fulfill $C^1$ contunity? How about $G^1$ continuity? (2p)

b) Demonstrate how Horner's rule can be used to accelerate the evaluation of polynomials. (2p)

c) One method to render text is to use a texture font, a font in a texture. Give one advantage and one disadvantage with this method. (2p)

## 6. Miscellaneous

a) Anti-aliasing can be performed by supersampling. Under what conditions (what kind of signal) will supersampling produce a good result, and why? (3p)

b) Given a point in a plane, describe how you can test whether that point is in a triangle in the same plane. (3p)

## 7. Collision detection and animation

a) Suggest a method to reduce the number of broad phase tests for scenes with a large number of objects. (2p)

b) Using a figure, give examples of two depth cues that are useful in 2D (pseudo-3D) graphics. (2p)

c) A simplified collision detection should be implemented, testing whether small objects (modeled as points) are inside a non-convex polyhedra. How can this be solved? Hint: The method in question was presented (mainly) as a 2D method in the lectures and the book. (2p)

## 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)