# EXAM IN

# COMPUTER GRAPHICS

# TSBK07

# (TEN1)

| | |
|---|---|
| Time: | 2nd of June, 2018, 14-18 |
| Room: | KÅRA |
| Teacher: | Ingemar Ragnemalm,<br>visits the exam around 15 and 17. |
| Allowed help: | None |
| Requirement to pass: | Grade 3: 21 points<br>Grade 4: 31 points<br>Grade 5: 41 points<br><br>ECTS:<br>C: 21 points<br>B: 31 points<br>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) When performing Phong shading, what information relevant to the shading is passed from the vertex to the fragment shader?

(1p)

b) The modifiers *in*, *out* and *uniform* are particularly important in GLSL. What do they stand for? Both the vertex and fragment shader should be taken into account.

(2p)

c) Describe how texture access is programmed. You may assume that you have a function that loads a texture to a texture object:

GLuint textureObject;
LoadTGA("mytexture.tga", &textureObject);

From this point, write code fragments (not a complete program) that shows how you can use this texture for rendering. Both CPU and GLSL code should be included.

Note: Exact names on functions or identifiers are *not* important, the task they perform is the important thing!

(3p)

## 2. Transformations

a) A matrix M is given:

$$M = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

plus a line passing through the point **p** in a direction given by a vector **v**. The matrix performs an operation along the **z** basis vector. Your task is to apply the operation along **v** using a sequence of matrix multiplications. The contents of each matrix should be specified. You do not have to multiply the matrices together.

(4p)

b) What does the matrix in (a) do? What does the 4's signify? (Hint: It is not just some scaling. It corresponds to something specific.)

(2p)

c) Two matrices are given:

$$S = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Describe what happens if you apply T*S or S*T, respectively, to a model. Is there any difference?

(1p)

**3. Light, radiosity and ray-tracing**

a) Describe Lambert's light model, that is, our model for Lambertian surfaces. For what part of the common 3-component light model does it apply?

(2p)

b) How are shadows produced in radiosity? Describe enough of the method to clarify how shadows are created.

(3p)

c) Describe how anti-aliasing is implemented in ray-tracing, as described in this course. There are two significantly different methods. How do the results differ? Motivate why one gives a better result than the other.

(2p)

**4. Surface detail**

a) Describe how *texture splatting* can be implemented. Your description should support at least four textures with a single splat map/blend map for full score. Code, pseudo code or formulas are expected in a full solution.

(3p)

b) Outline how *environment mapping* works. Significant underlying techniques (beyond standard operations like texture mapping) should also be described.

(2p)

**5. Curve generation**

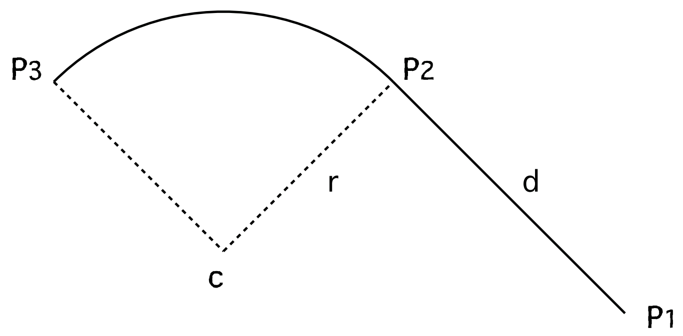
a) Describe how surfaces can be built from splines. Which spline or function is your solution based on?

(2p)

b) Is a Bézier curve an approximation or interpolation spline? Motivate your answer. (A relevant figure is recommended.)

(2p)

c) 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 $\mathbf{c}$, 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) Describe how a geometric, self-similar fractal works (e.g. like the Koch curve or a simple tree). The description should outline the algorithm in sufficient detail as pseudocode.

(2p)

b) Describe how flood fill can be implemented (the basic version). Your solution should be significantly more efficient than a simple recursion pixel by pixel. Figures are recommended.

(3p)

c) Describe, using a figure, the difference between the odd-even rule and the non-zero winding number for inside-outside tests.

(1p)

**7. Collision detection and animation**

a) Describe, with brief text and figures, two pseudo-3D effects that are useful for 2D animations.

(2p)

b) Describe the broad phase-narrow phase model for collision detection. There are really three phases. Which ones? Suggest a possible solution for each phase. Descriptive name or short description, possibly a figure, will suffice.

(3p)

**8. Visible surface detection and large worlds**

a) 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)

b) Describe how view plane oriented billboards can be implemented for a billboard located at the point **p**, using a model-to-view matrix M.

(2p)