EXAM IN

COMPUTER GRAPHICS

TSBK07

(TEN1)

Time:	3rd of June, 2023, 14-18
Room:	TER3 & TERE
Teacher:	Ingemar Ragnemalm, visits the exam around 15 and 17, 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) Give two examples of tasks that a vertex shader often does other than transform the location of vertices.

(2p)

b) A texture is prepared for access by a shader like this:

```
GLuint tex;
LoadTGATextureSimple(”myTexture.tga”, &tex);
glUniform1f(GetUniformLocation(”tex”), tex);
```

where LoadTGATextureSimple (straight from the lab) loads the texture from disk and uploads to the GPU.

But… Not exactly. What is the correct method? Why is this wrong?

(3p)

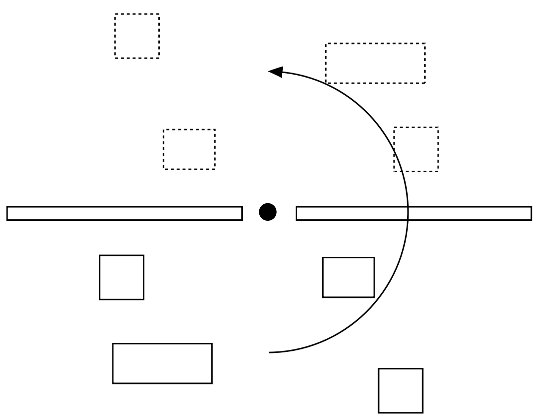
2. Transformations

a) In lab 2, a student realises that the bunny had to be moved to get in view when a projection transform was applied. The student tries to translate along Z by -1, but accidentally uploads it transposed! What operation did the student actually perform?

Write the matrix in question and explain what it is doing in its transposed version.

(2p)

b) In my ”The Wall” demo (do you remember?) I created a simple portal effect by rotating the entire scene around the portal.



Given the portal center as the point **p**, and a rotation by π, give a sequence of matrices to produce this effect. The contents of each matrix should be given. You do not have to multiply the matrices together.

(3p)

c) What statements are correct? R_z = rotation around Z, S = scaling, T = translation, M^{-1} = inverse, M^T = transpose. All correctly stated as correct or not = 2p, one error = 1p.

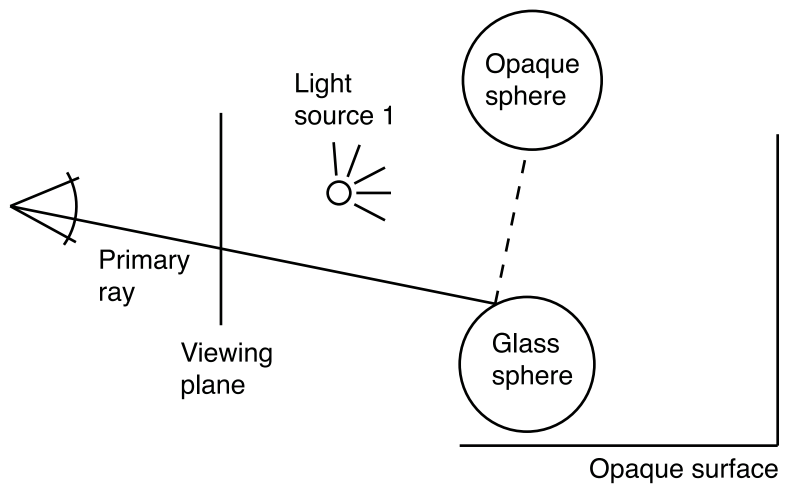
- $S(2, 2, 2) * R_Z(\pi/2) = R_Z(\pi/2) * S(2, 2, 2)$
- $T(0, 2, 0) * R_Z(\pi/2) = R_Z(\pi/2) * T(2, 0, 0)$
- $S^T(2, 2, 2) = S^{-1}(2, 2, 2)$
- $R_Z^T(\pi/4) = R_Z(-\pi/4)$

(2p)

3. Light, shading and ray-tracing

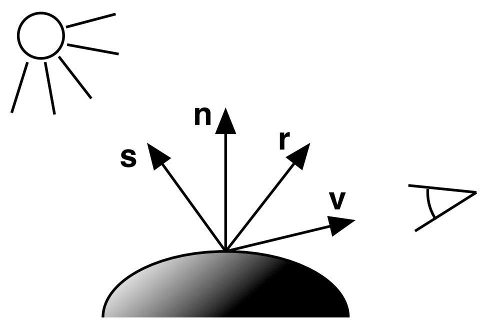
a) A primary ray, show in the figure below together with one help line and a few surfaces, is produced as part of a ray-traced rendering. All objects are in the same plane (e.g. your paper). The opaque surfaces do not cause any reflection or refraction while the glass sphere only creates these. Describe all rays that have to be traced from this primary ray. No jittering is involved.

(3p)



b) Write a formula for the 3-component light model allowing for multiple light sources. Use the symbols of the following figure:

(3p)



4. Surface detail

a) Spherical coordinates (q, v) can be defined by

$$x = R\cos(\boldsymbol{\varphi})\cos(\theta)$$
$$y = R\cos(\boldsymbol{\varphi})\sin(\theta)$$
$$z = R\sin(\boldsymbol{\varphi})$$

Write formulas for spherical texture mapping, mapping x, y, z to texture coordinates (s, t), normalized to the interval [0, 1].

Mathematical functions like $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$ may be used. If calls outside the usual math theory is used, their function (like output range) must be clearly stated.

(3p)

b) When using mip-mapping, it is often described as *tri-linear interpolation*. Why is it tri-linear? What is interpolated?

(2p)

c) Describe how to perform *splatting* (multi-texturing with a blend map). How many textures can you use with your solution?

(2p)

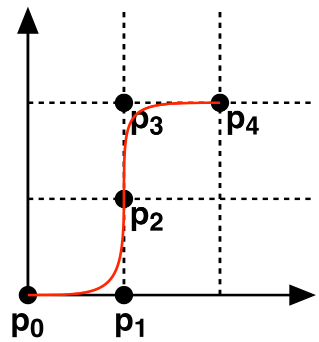
5. Curve generation

a) A quadratic Bézier spine is described by the following equations:

$\mathbf{p}(u) = (1-u)^2\mathbf{p}_0 + 2u(1-u)\mathbf{p}_1 + u^2\mathbf{p}_2$

Five points are given as

$\mathbf{p}_0 = (0,0)$
$\mathbf{p}_1 = (1,0)$
$\mathbf{p}_2 = (1,1)$
$\mathbf{p}_3 = (1,2)$
$\mathbf{p}_4 = (2,2)$



Two quadratic Bézier curves are drawn using $\mathbf{p}_0\ \mathbf{p}_1\ \mathbf{p}_2$ and $\mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4$. Prove that these curves meet with G^1 continuity! A mathematical proof is needed for full score.

(4p)

b) Is the Bézier curve approximating or interpolating? How can you tell that from the blending functions?

(2p)

6. Miscellaneous

a) Describe how the fractal ”Koch curve” (”snowflake”) is produced. Geometric shapes and necessary mathematical operations should be included.

(3p)

b) Anti-aliasing can be performed by *supersampling*. For simplicity, do this analysis in 1D. If 4x supersampling is used, how much can you expect aliasing to be suppressed? The explanation should involve arguments in frequency space.

(3p)

7. Collision detection and animation

a) Give an example of how the *global phase* (as defined in the course) can be implemented.

(2p)

b) A simplified case for collision detection is *sphere vs polyhedra collisions*. Describe how you can do sphere-polyhedra collision detection, and argue why/if some tests can be skipped to simplify the tests (possibly when using it for polyhedra vs camera collision detection).

(3p)

c) Collision handling is particularly easy when handling an object (sphere) colliding with a stationary object like the ground, i.e. an object which we consider having infinite weight. How can you handle such a collision? The solution should include the restitution parameter ε, the elasticity of the collision.

(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? The description should reflect the coordinate systems involved.

(3p)

b) How can you implement view plane oriented billboards?

(2p)

c) Back-face culling is not done using the normal vectors provided by the model. Why?

(1p)