EXAM IN

COMPUTER GRAPHICS

TSBK07

Time:	30th of May, 2012, 8-12
Room:	TER4
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

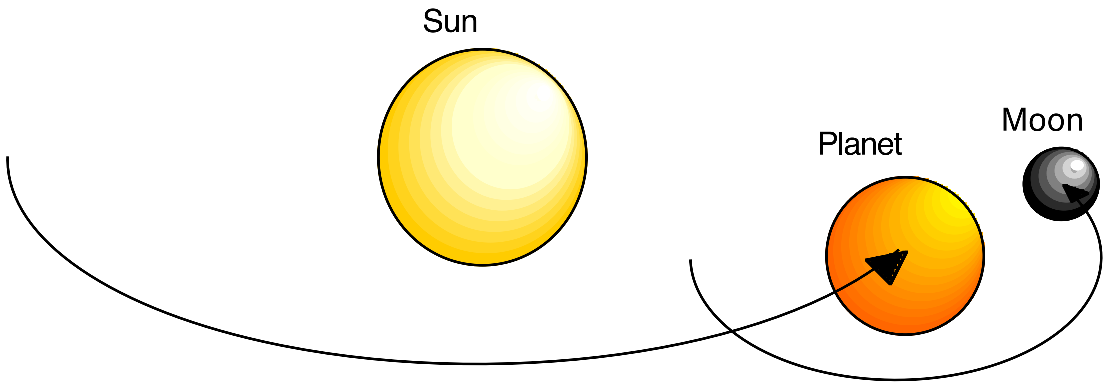
Answers may be given in swedish or english.

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) The figure shows a small planetary system consisting of one sun, one planet and one moon. The planet rotates around the sun, and the moon around the planet, all in the XZ plane.



The sun is drawn at x_S, y_S, z_S, the planet rotates around the sun at radius r_p, with the angle α, and also rotates around its own Y axis by the angle β. Similarily, the moon rotates around the planet at radius r_m, angle χ, and rotates around its Y axis by angle δ.

Function calls drawSun() , drawPlanet() and drawMoon() are provided, all drawing the object centered around origin. Outline how an OpenGL program would draw the planetary system.

You may make reasonable assumptions on the availability of a suitable math library. Fragment shader should not be included, but any relevant mathematical operations in the vertex shader should be described. (4p)

b) For many graphics effects, data need to be interpolated from one vertex to another. Describe how this works in a shader program by writing a simple example. Any kind of data may be chosen for the example. (3p)

2. Transformations

a) A rigid model should be drawn in a scene. It may be placed anywhere in the scene with any orientation. Give a typical sequence of transformations for performing this. *All transformations* that need to be specified in an OpenGL program should be included, and the sequence should be suitable for intuitive adjustment to fine-tune the placement of the model in the world.

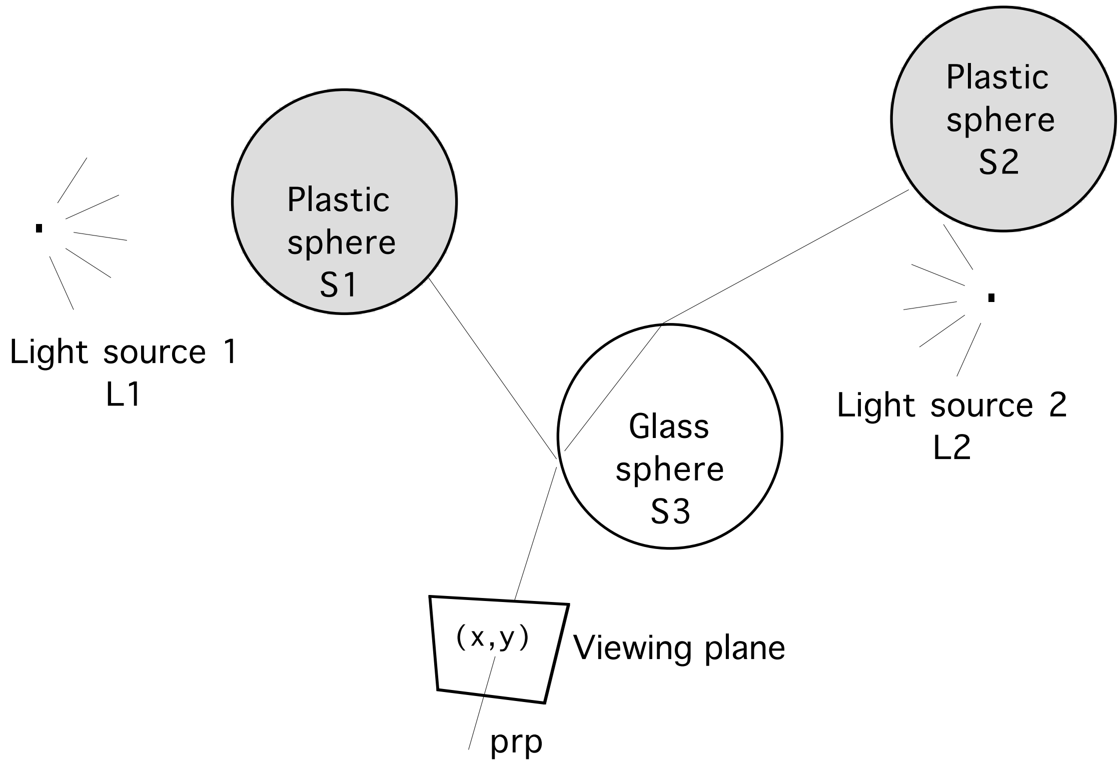
Standard transformations (translation, rotation...) matrices only need to be given by name (their contents are relevant in part b), but special matrices should be given in detail. You do not have to prove that they are correct, and simplified versions can suffice when appropriate. Description without contents will give partial score.

(4p)

b) Two points, **p1** and **p2**, located in the same XZ plane (same Y component), define an axis in 3D space. Produce a sequence of 4x4 matrixes that define a transformation that performs rotation around this axis by an angle ϕ. The contents of each matrix should be given. You don't have to multiply the matrices together.

(4p)

3. Light, shading and ray-tracing



The model above is rendered using ray-tracing. One particular primary ray is shown, plus the resulting secondary rays. L1 and L2 are point sources. S3 is a perfect glass sphere, with only reflection and refraction.

a) For the plastic spheres, the three-component light model should be used. Write a formula describing this model. A figure for defining the geometry is recommended.

(3p)

b) Outline a ray-tracing algorithm that can produce the shown rays and calculate a pixel value. (That is, a pseudocode description of the main function, with suitable subroutines given with descriptive names only.)

(3p)

c) For the plastic spheres, how do you determine what light sources to use in the calculation of the light level?

(2p)

4. Miscellaneous

a) Mandelbrot, the Koch curve and fractal plants are all extremely different data, but they all share some basic features that all fractals have. What features are these?

(2p)

b) Explain, preferrably using reasoning in the frequency domain, why removing aliasing by (linear) post-filtering does not perform as well as supersampling.

(2p)

5. Surface detail

a) Cylindrical coordinates (θ, v) can be defined by

$$x = R \cos\theta$$
$$y = R \sin\theta$$
$$z = v$$

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

Only single-argument mathematical functions like $\tan^{-1}$ may be used.

(3p)

b) If you calculate texture coordinates as in a), why can't you just map texture coordinates to each vertex? What error will you get and how do you fix it?

(2p)

6. Curve generation

a) Use Bresenham's line drawing algorithm to draw a line from (5, 5) to (10, 8). Describe step by step how this works, using a figure and a table showing how variables change.

Hint: $p_0 = 2\Delta y - \Delta x$ (4p)

b) A quadratic Bézier (three-point Bézier) segment can be written

$$P(u) = (1 - u)^2 P_0 + 2(1-u)uP_1 + u^2 P_2,\ u \in [0, 1]$$

(This is not the Bézier in the book, but similar and the same methods apply.)

Show that two adjacent segments (P0-P1-P2 and P2-P3-P4) will be continuous if P1, P2 and P3 are on a line. Under what circumstances do you get G^1 or C^1 continuity?

(4p)

7. Collision detection and animation

a) Suggest a bounding shape that will make broad phase tests really simple. Explain why this is not necessarily a good solution.

(2p)

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.

(3p)

8. Visible surface detection and large worlds

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

(2p)

b) Describe and compare two large world VSD methods. You may choose any two methods described during the course. At least two qualitative differences should be described.

(3p)