EXAM IN

COMPUTER GRAPHICS

TSBK07

Time:	11th of August, 2008, 8-12
Room:	Ter1
Teacher:	Ingemar Ragnemalm, visits around 10
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.

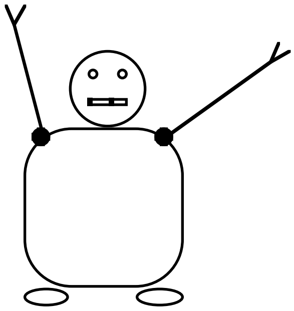
Good luck!

1. OpenGL

a) The modelview matrix is initialized to the identity matrix. Give a sequence of transformation calls (glTranslatef and glRotatef) that place the camera in (2, 0, 2) looking at origin.

(3p)

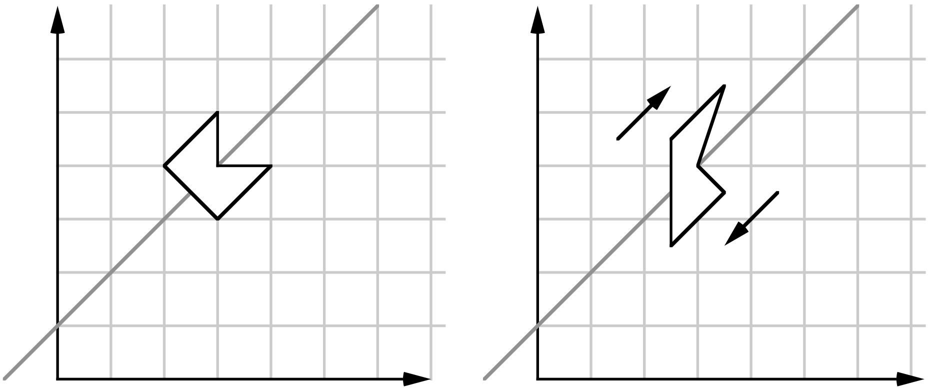
b) Demonstrate how the calls glPushMatrix() and glPopMatrix() can be used by writing the most relevant code for drawing the robot in the figure below. It has two moveable arms, using separate rigid bodies for the arms. You do not need to give arguments to transformation calls, and may abstract other parts (e.g. geometry) as needed.



(3p)

2. Transformations (etc)

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. You don't have to multiply the matrices together.



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

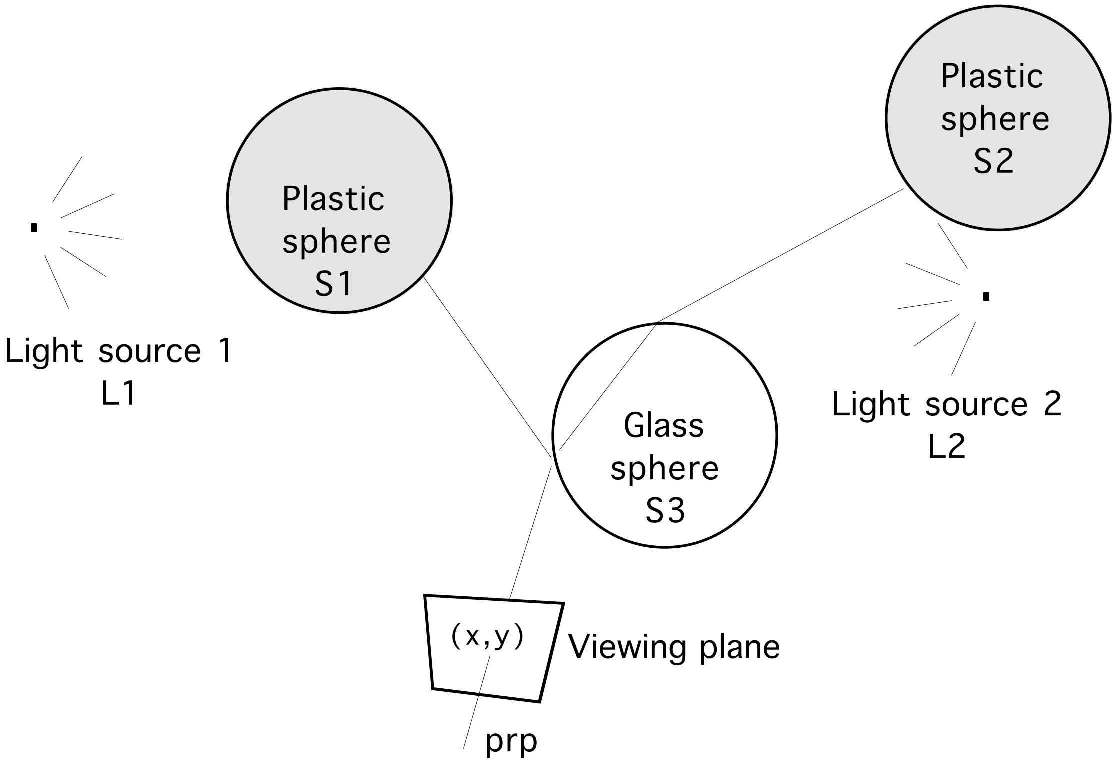
After transformation

(4p)

b) Under what condition is it possible to invert a matrix simply by transposing it? For what geometric transformations does that apply?

(2p)

3. Light, shading and ray-tracing



a) 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.

What components in the model contribute to the resulting intensity in x,y? What calculations are used to decide which parts contribute and which ones do not?

(3p)

b) When a ray hits a transparent surface, the refracted ray needs to be calculated. Outline, using a figure, the geometrical principle for the solution. You do not need to give a full mathematical derivation.

(3p)

4. Mapping techniques

a) Describe, with formulas, how to map a texture onto an object using a cylinder as intermediate surface.

(4p)

b) Explain how mip mapping works. Why doesn't mip mapping always produce perfect results?

(2p)

5. Shaders

Below follows a few lines of GLSL code that I hope not to see written in the lab. Not only is the code incomplete and rather meaningless (except for the purpose of amusing the examiner), but there are some details that may not work as expected even in code intended to do meaningful work.

```
#include ”glsl.h”
int main()
{
      char *buffer;
      vec3 color = new ColorObject(1, 2, 3);
      uniform float gl_time;
      varying shade;
      color.rgb = color.xyz;
      color = color + gl_Vertex;
      if (color.r = 0)
            printf(”Red is zero\n”);
      gl_time += 1;
      *buffer = gl_time;
      color.r = clamp(gl_time, 0, 1);
      shade = gl_Normal.zzz;
      gl_Position = gl_ProjectionMatrix + glModelViewMatrix / gl_Vertex;
}
```

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.

(6p)

6. Fundamental 3D geometry for graphics

A triangle is given by three points, **a**, **b**, **c**, and a line segment is given by $\mathbf{d} + \mu * \mathbf{e}$, $0<\mu<1$.

a) How do you calculate the normal vector of the plane that the triangle is located in?

(2p)

b) How do you calculate if and where the line segment intersects the plane?

(2p)

c) How do you determine if that point is inside the triangle or not?

(2p)

7. Collision detection and large worlds

a) Suggest three bounding shapes that are suitable for the broad phase.

(2p)

b) Describe how 1-dimensional sorting can be used for reducing the number of collision detection tests in a scene with a large number of objects.

(2p)

c) How does geomipmapping work? Outline the principles for the method with a figure. How do we select and switch between resolutions?

(3p)

8. Visible surface detection

a) How do you construct a BSP tree from a set of polygons? Outline the algorithm using understandable pseudo code.

(3p)

b) Discuss advantages and disadvantages of BSP trees compared to the Z-buffer method. You should be able to give some advantage of each.

(2p)

c) Describe a method for backface culling (back-face detection).

(2p)