EXAM IN

COMPUTER GRAPHICS

TSBK05

Time:	11th of March, 2005, 14-18
Room:	TER2
Teacher:	Ingemar Ragnemalm, visits around 15
Allowed help:	None
Requirement to pass:	Grade 3: 21 points Grade 4: 31 points Grade 5: 41 points C program: Usually numbers as above, but if you like, we can use the following scale: G: 21 points. VG: 36 points. ECTS: E: 21 points D: 26 points C: 31 points B: 36 points A: 41 points

Answers may be given in swedish or english.

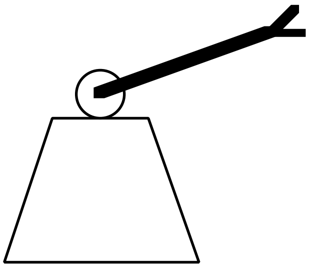
Note: Students who have followed TSEA55 are welcome, but should clearly state that the result should be registered for TSEA55 instead of TSBK05.

Good luck!

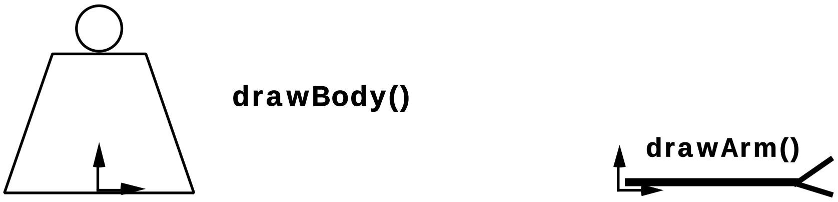
1. OpenGL programming

a) Suppose you are making an OpenGL application that moves the camera every frame. The routine display() is called repeatedly, while the routine reshape() is only called when the window is created or its size changes. The camera is managed by the calls glFrustum(), gluLookAt(), glPushMatrix(), glPopMatrix, glMatrixMode() and glLoadIdentity(). Outline how these calls are used, in what order and in what routines.

b) A robot model with a simple arm (one joint only) is modelled centered around origin. With the calls drawBody() and drawArm() you can draw its parts. See figures below.



Robot with one arm



Robot body and arm, given as function calls, with origin.

Write OpenGL code for placing the body and the arm in the world, and rotating the arm in one dimension. The joint is placed directly above the body’s origin, on the distance D. You should not write code for the entire program, only for the parts that deals with the drawing of the robot.

(4p)

2. Curve generation

a) A parabola is given by the formula $y = x^2 / 16$. Derive the incremental updating of the decision variable in the midpoint algorithm to plot this curve, 8-connected, starting at (0,0) and continuing in positive x direction. Also, calculate the starting value for the decision parameter.

b) In order to draw any part of the curve in a), you must use different expressions for different parts of the curve. At what points must the algorithm switch expressions, and why? Illustrate this in a figure. (You don't have to derive the incremental updating expressions for the other parts.)

(6p)

3. Transformations

a) In 3D space, a plane is given by a point in the plane P and a normal vector N. The vector N is normalized. P and N may vary, but we know that N is always in the x-z-plane, so the y component is always zero. We wish to perform a mirroring transformation over the plane defined by N and P.

Give a sequence of 4x4 matrixes, each defining one basic geometric transformation (translations, rotations, scaling, mirroring and skewing), that together define a transformation that performs mirroring over the given plane. Derive the contents of each matrix as numeric values or appropriate, clearly defined symbols. Write the total operation as a sequence of matrix multiplications. You don't have to multiply the matrices together.

b) The following matrix is a projection matrix.

$$\begin{matrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{matrix}$$

The parameters a and b may seem meaningless, since they do not affect the projected x and y values. How can they still be of any use? How are they chosen? (You do not have to derive their exact values.)

(6p)

4. Rendering and surface mappings

a) What is mipmapping? How much extra memory does mipmapping require?

b) 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].

c) Describe the Floyd-Steinberg algorithm for halftoning.

(8p)

5. Light, shading and ray-tracing

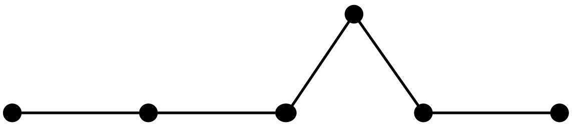
a) Describe how anti-aliasing is performed in ray-tracing.

b) Motivate why radiosity generally only supports Lambertian surfaces. What parts of the radiosity light model has problems modelling specular surfaces?

(4p)

6. Miscellaneous

a) A fractal similar to the Koch curve was generated by the generator in the figure below and a square as initiator. What is the fractal dimension of this fractal? (Hint: $ns^D = 1$)



b) Describe how Flood Fill is implemented. Recursive algorithms with high recursion depths are not acceptable.

c) Using a figure, illustrate the difference between the odd-even rule and the non-zero winding number rule. The figure should have at least one area where the two rules give different result.

(7p)

7. Collision detection

a) Describe, in reasonable detail, how do you detect whether a line intersects a triangle. Both the intersection with the plane and the check for whether the intersection is inside the triangle should be detailed. The triangle is given as three points, A, B and C.

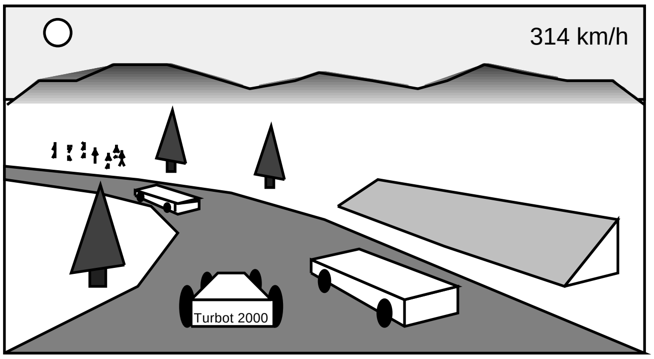
b) Describe, using a figure, how camera-polygon collisions are resolved.

c) When objects move fast, they run a risk to pass straight through each other without detecting a collision. How can the collision be properly detected?

(8p)

8. Visible surface detection and scene management

a) Linus and Linnea have created a particularly pleasing project, a car racing game, illustrated by the figure below, although their scenes are much more detailed. In fact, you should consider their scene to be extremely detailed! The scene includes the cars, the racing track, trees, sky, surrounding terrain, etc.



The game was implemented with OpenGL, rendering all polygons with the GPU. Despite the relatively fast GPU, they have performance problems. They have found that the problem is that the polygon count is too high for the GPU. They need to render the scene in not more than half the time. They do not want to sacrifice more visual quality than necessary. Removing texture mapping for any surface is not an option, neither is lower screen resolution. The loss of visual quality should be almost unnoticable.

Suggest a remedy that would compromise rendering speed and rendering quality for this application. Hint: Different parts of the scene can be handled in different ways. Describe what parts should be rendered one way or the other.

b) Write pseudo code for how to generate a BSP tree from a set of polygons.

c) It is possible to use the same BSP tree for speed optimizations, for avoiding to process parts of the scene that are not visible anyway. How? Illustrate your answer with a figure.

(7p)