

Exam in Computer graphics

1TD381;1TT180

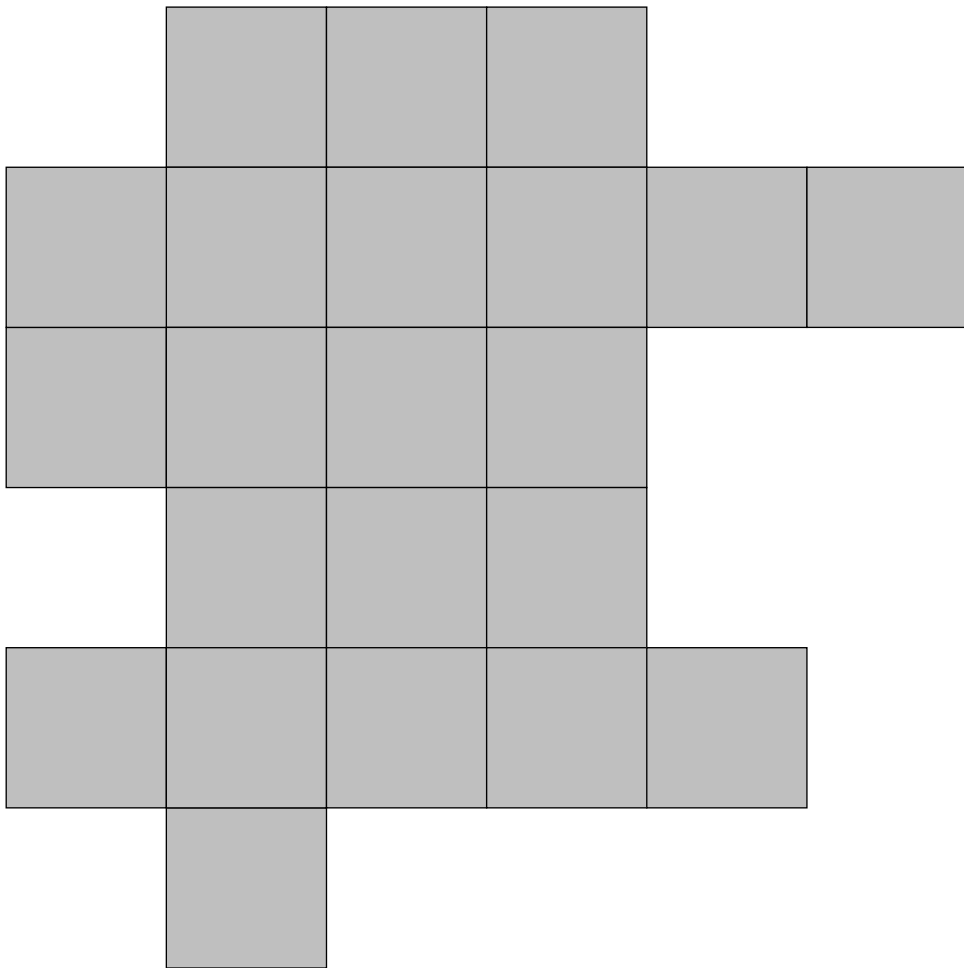
Time and place: 08:00–13:00 Friday June 3 2005, Room 6140 (MIC-klubben), Polacksbacken.

Instructions: Check that you have received the correct exam. The cover sheet shall always be filled in and returned even if no exam questions have been answered. Only paper, pen(cil), erasers, rulers, and dictionaries are allowed. No calculators allowed. Write your name on all papers that you hand in. Do not use red ink. Write only on one side of each paper. Start on a new paper for each question. Sort the answer sheets in the order of the questions before you turn them in.

Grades 1TD381: Pass (G): 20pts; Pass with distinction (VG): 30pts.

Grades 1TT180: 3: 20pts; 4: 26pts; 5: 34pts.

Figure for question nr 5



1 Rasterization

Ingela

Very basic tools in Computer Graphics are algorithms to rasterize a line segment defined by two vertices, i.e., its start-point and end-point.

- (a) Describe a straight-forward way to set the pixels between two points best corresponding to a straight line. Here, floating-point arithmetic is allowed. (2p)
- (b) Bresenham derived a line-rasterization algorithm that avoids floating-point arithmetic. Describe this standard approach used in hardware and software rasterizers. (3p)

2 Transformations, viewing, and homogeneous coordinates

Patrick

- (a) Draw a view frustum. Position, and name, the three important rectangular planes at their correct positions. Make sure that the position of the origin, and the orientation of the z -axis is clearly distinguishable. In what coordinate system is the view frustum defined? (2p)
- (b) Two of the previous planes define a view volume. Only objects within a view volume may be visible to the observer. Derive the formula for the perspective projection (from 3D to 2D) of a point \mathbf{p} at position (x, y, z) within the view volume, onto the third plane (the plane closest to the observer). Draw an illustration. The distance from the observer to the plane closest to the observer is d . Finally express the 3D homogeneous coordinate perspective projection matrix \mathbf{P} . (3p)

3 Colour

Ingela

- (a) The colour we perceive is a combined result of some different "parts" the light information passes on the way to the brain. What are the different subparts that affects the perceived colour of an object? (2p)
- (b) Give a motivation why we usually use a 3D space in order to represent colour. (1p)
- (c) How are the two colour models RGB and CMY related to each other? How is the colour system HLS constructed? Explain the letters by words and illustrate. (2p)

4 The Phong Reflection Model

Ingela

The Phong reflection model is an approximation of the physical reality to produce good renderings under a variety of lighting conditions and material properties.

- (a) Describe the four vectors the model uses to calculate a colour for an arbitrary point p . Illustrate with a figure. (2p)
- (b) What are ambient, diffuse, and specular reflection, respectively? (3p)

5 Visualization

Ingela

- (a) Imagine a discrete 2D image consisting of pixels being either object (1) or background (0). One way to represent the object is by its contour. One method for this is called *marching squares*. How does it work? What is the contour by marching squares of the shape on the backside of the first exam sheet? Hand this sheet in if you do not want to draw your own figure. (2p)
- (b) If we extend the marching squares argument to surfaces in three dimensions, we get a method called *marching cubes*. Describe the six basic steps of this isosurface construction method. (3p)

6 Curves and Surfaces

Erik

- (a) Consider the second degree polynomial $p(u) = c_0 + c_1u + c_2u^2 = \mathbf{u}^T \mathbf{c} = [1 \ u \ u^2][c_0 \ c_1 \ c_2]^T$ and the control points $\mathbf{p} = [p_0 \ p_1 \ p_2]^T$. Derive the matrix A so that the coefficients $\mathbf{c} = A^{-1}\mathbf{p}$ make $p(u)$ interpolate all control points (for $u \in [0, 1]$). (3p)
- (b) Bezier polynomials can be rendered efficiently with fast evaluators or recursive subdivision. It is common to convert a non-Bezier polynomial (e.g., the one in (a)) to an equivalent Bezier polynomial in order to use these rendering techniques. Describe algebraically how to do this. (2p)

7 Mapping

Anders

- (a) Which are the main principles behind environment mapping and what is the visual effect? (2p)
- (b) What is the general idea behind bump mapping? Also explain what the map itself contains. (3p)

8 Hierarchical Modeling and OpenGL

Erik

Consider the planets A and B in the figure below. Their radii are R_A and R_B , respectively. A is positioned at $(0, 0, 0)$ and B is positioned at $(d, 0, 0)$ initially. From a simulation we get the angles θ_A (rotation of A about its own y -axis), θ_B (rotation of B about its own y -axis), and θ_{BA} (rotation of B about A 's y -axis). Your task is to write OpenGL pseudo-code to visualize this system. To draw the planets you have the function `drawUnitSphere()` that draws a sphere with radius 1.0 centered at $(0, 0, 0)$. The other functions you may need are:

(5p)

- `glRotate(angle, axisx, axisy, axisz)`
- `glTranslate(dx, dy, dz)`
- `glScale(sx, sy, sz)`
- `glPushMatrix()`
- `glPopMatrix()`

You only need to write pseudo-code for transformations and drawing.

