TDA362/DIT223 Computer Graphics

EXAM

(Same exam for both CTH- and GU students)

Friday, January 18th, 2019, 08:30 - 12:30

Examiner
Ulf Assarsson, tel. 031-772 1775, 072-974 4817.
Questions during exam: Dan Dolonius: 0707-76 91 78.

Permitted Technical Aids
None, except English dictionary

General Information
Numbers within parentheses states the maximum given credit points for the task. Solutions shall be clear and readable. Too complex solutions can cause reductions in the provided number of points

Questions to examiner during exam
will be possible approximately one hour after the start of the exam. If anything is unclear – remember what has been mentioned on the lectures, in the slides and course book and do your best.

Grades
In order to pass the course, passed exam + exercises (or exam + project) are required. The final grade is calculated from the exam grade. The exam is graded as follows

CTH: 24p ≤ **grade 3** < 36p ≤ **grade 4** < 48p ≤ **grade 5**

GU: 24p ≤ **G** < 45p ≤ **VG**

Max 60p

Grades are announced by the LADOK system ~3 weeks after the exam

Solutions
will be announced on the course home page.

Review
Review date (granskningsdatum) will be announced on the course home page.

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

Question 1

a) **[2p]** What is the ModelViewProjectionMatrix? What does it do?
Answer: The matrix that transforms from modelspace (via worldspace) into viewspace and finally into unit- or projection space (or normalized device coordinates or homogeneous coordinates).

b) **[2p]** Explain screen tearing.
Answer: see lecture 1. Solved by double buffering and syncing buffer swapping with the vblank.

c) **[1p]** What is the advantage of Bresenham's line drawing algorithm?
Answer: uses only integers

d) **[1p]** Assume that **M** is a matrix that is used to transform triangle-vertices and that **M** not necessarily is orthogonal. What is the matrix to correctly transform the normals?
Answer: $(\mathbf{M}^{-1})^{T}$

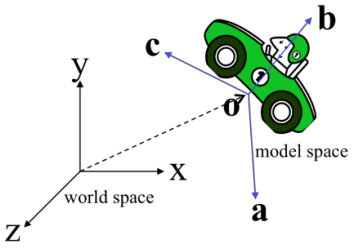
e) **[3p]** State which matrix components control 1) **scaling** in **x-**, **y-**, and **z**-direction, 2) **rotations** in general, and 3) **translation** in the **x-**, **y-**, and **z**-direction?

$$\begin{bmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \\ d & h & l & p \end{bmatrix}$$

Answer:
rotation: abcefgijk.
scaling: x: a, y: f, z: k
translation: x: m, y: n, z: o

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

f) **[1p]** State the object's model-to-world matrix.



Answer: $\mathbf{M}_{\text{model-to-world}} = \begin{bmatrix} a_x & b_x & c_x & o_x \\ a_y & b_y & c_y & o_y \\ a_z & b_z & c_z & o_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

Question 2

a) **[2p]** Give two reasons for gamma correction.
Answer: 1) screen output is non-linear so we need gamma to counter that. 2) Textures/images can be stored with better precision (for human eye) for low-intensity regions.

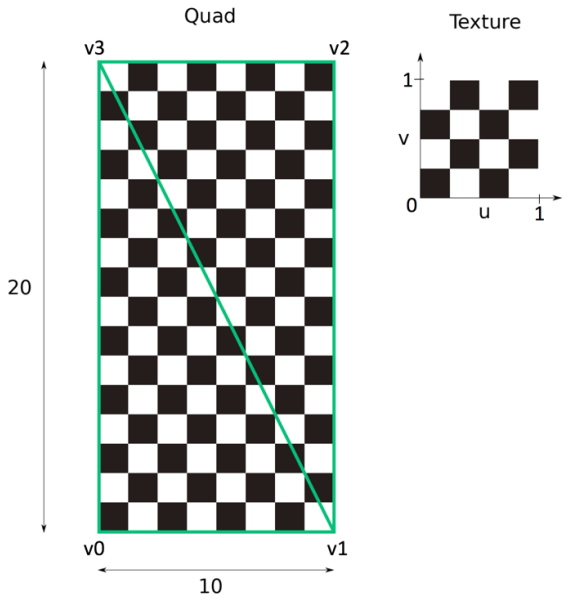
b) **[1p]** Does the rendering order of transparent objects (e.g., triangles) matter? Motivate for any point.
Answer: Yes. Back-to-front order (see lecture 3).

c) **[1p]** Describe how anisotropic filtering works. Also tell why it is needed.
Answer: Up to for instance 16 mipmap lookups along the line of anisotropy.
A different amount of filtering in the *x* and *y* direction is generally needed.

d) **[2p]** What is 2x2 grid, 2x2 RGSS, and 8 rooks. Draw them!
Answer:

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

e) **[2p]** Assume you want to texture a quad with a repeating chessboard pattern (see the figure). You start by setting the texture wrap option to REPEAT. What texture coordinates (*u,v*) should you specify for each vertex v0,v1,v2,v4 to achieve the desired effect?
Answer: v0:(0,0), v1:(2,0), v2: (2,4), v3:(0,4)



f) **[2p]** Assume that you are rendering an impostor (i.e, billboard) as a rectangular quad. The impostor's texture represents a tree and contains lots of fully transparent texels. How do you assure that objects that are later (for the same frame) rasterized behind the transparent parts of the impostor will show on the screen? (**Hint:** use the fragment shader.)
Answer: alpha test, fragment kill, i.e., kill fragment if alpha = 0 in fragment shader.

Question 3

a) **[3p]** What is a vertex shader, geometry shader and fragment shader? Explain their functions, respectively.
Answer:
Vertex shader: per vertex operations like projection to unit-cube (2D) and per-vertex attributes to be interpolated (color,normal).
geometry shader: take input primitives and output possibly another amount and possibly another type of primitive.
Fragment shader: output pixel color from interpolated data from vertex shader.

b) **[1p]** Normally, you would want to draw all geometry that is in front of the camera. So, why is a near and far plane used for the view frustum?
Answer: Near plane is used to avoid a degenerate projection plane. Far plane can be used to get better z-precision in the z buffer.

c) **[4p]** Describe a method for a ray-polygon intersection test. (You may assume that the polygon is convex and lies in a 3D plane.)
Answer: See lecture Intersections: E.g., convert to a point-in-2D-polygon test and use, for instance, the *crossing number algorithm / even-odd rule*. (Answers based on splitting into triangles and using analytical ray-triangle intersection are OK. There are also many other alternatives that are correct.)

d) **[1p]** Describe a test which determines whether or not two spheres intersect.
Answer: $||c_1 - c_2|| <= r1 + r2$.

e) **[1p]** Describe a test which determines whether or not a sphere and a plane intersect.
Answer: Insert the sphere center into the plane equation. If the result is smaller than the radius, there is an intersection.
I.e., $||\mathbf{n} \bullet \mathbf{c} + d|| <= r$.

Question 4

a) **[4p]** Describe how to sort objects (roughly) in back-to-front order when they are stored in an Axis Aligned BSP-tree.
Answer:
At each node:
if leaf, then draw the object.
Recursively continue traversal for the child at the further side of the node-plane.
Recursively continue traversal for the child at the hither side of the node-plane.

b) **[1p]** Name a sorting algorithm that is efficient when we have high frame-to-frame coherency (regarding the objects to be sorted per frame)?
Answer: Bubble-sort (or insertion sort), which have expected runtime of resorting already almost sorted input in O(1) instead of O(n log n), where n is number of elements.

c) **[1p]** How do you nicely terminate recursion in ray tracing if you do not simply want to terminate at a maximum recursion depth?
Answer: terminate when the ray's influence on the final pixel color is below a certain threshold. (Send a weight with the reflection/refraction rays.)

d) **[2p]** Describe a common recursive pattern for adaptive super sampling (suggestively the one that was taught on the lectures). Draw start samples, samples after one level of recursion and criteria to terminate or continue the recursion.
Answer: see lecture Raytracing 1

e) **[2p]** Explain how a skippointer tree works and what the advantage is?
Answer: The (BVH)-tree stored in depth first traversal order, sequentially in an array, with a pointer/index to the next element if traversal of subgraph should be skipped. **Advantage:** gives good cache coherence.

Question 5

a) **[3p]** Explain Final Gather and how it is used with the photon maps. Also, explain what is good with Final Gather.
Answer: Monte Carlo-sample at first bounce using many stochastic rays. (Use caustics map at first intersection and global+caustics) photon map at intersections of secondary rays. Good because lowers noise from the global map. (No reduction if caustics and global map are not specified.)

b) **[1p]** How are soft shadows from an area-light source computed with **path tracing**?
Answer: At each intersection, one shadow ray is shot to **one** random position on the light source.

c) **[1p]** Which properties of a material are described by the Fresnell effect?
Answer: degree of transmission and reflection.

d) **[2p]** Describe the advantages and disadvantages of shadow maps vs shadow volumes. You should mention at least a total of four important bullets (0.5p per unique bullet).
Answer: SM Pros: any rasterizable geometry, constant cost per rasterized fragment from the camera's view (basically just a texture lookup), fast.
SM Cons: jagged shadows / resolution problems, biasing.
SV Pros: sharp shadows.
SV Cons: 3 or 4 rendering passes (and thus often slower than shadow maps), lots of polygons and fill.

e) **[2p]** Why do you need to use a bias in the shadow map algorithm? What is the cause of the problem?
Answer: We compare two different discretizations - one from the eye and one from the camera. (To avoid z-fighting (incorrect self shadowing) and light leaking.) The view sample can lie further from the light than the shadow map sample (due to discrete sampling).

f) **[1p]** What is good with the z-fail algorithm, compared to z-pass?
Answer: avoids the eye inside shadow problem

Question 6

a) **[1p]** Sketch one non-continuous curve and one C^0-continuous curve (and mark which is which).
Answer:

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

b) **[2p]** Explain what C^k-continuity means and what G^1-continuity means for curves.
Answer: C^k-continuity means that the derivatives of order 0 (or 1) to k exists and are continuous. G^1-continuity means that the tangent vectors between each curve segment have equal directions, but their lengths (strengths) may vary.

c) **[1p]** What is the potential advantage and disadvantage of Bezier-curves compared to Hermite curves? (Think of a user that has to specify the curves)
Answer: Bezier - no need to give tangent vectors manually
Hermite curves - can guarantee continuous 1st derivatives between segments.

d) **[1p]** Assume **p**=(0,8,6,4). Perform the homogenisation step on **p**.
Answer:

e) **[1p]** Given two vectors ω_i and ω_o, how does one calculate the half angle vector ω_h?
Answer: $\omega_h = \text{normalize}(\omega_i + \omega_o)$.

f) **[4p]** Draw the graphics-pipeline's major functional blocks (i.e., logical layout) and their relation to hardware, for a modern graphics card. (Hint: preferably I want the major functional blocks as described in the hardware lecture, e.g. with the different parallel shader units and the other functional units).
Answer:

Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY