

MIDTERM EXAMINATION

CO19-320322: COMPUTER GRAPHICS
320322: GRAPHICS AND VISUALIZATION

Fall 2016

Prof. Dr. Lars Linsen
Jacobs University Bremen
October 31, 2016

Name: _____

Signature: _____

Problem	1	2	3	4	5	Σ
Points						
Max. points	7	15	14	15	14	65

Notes:

- Explain all computation steps for all your answers!!!
- No books, notes, or calculators may be used.
- Work on your own without copying or cheating.
- Hand in all your paperwork and clearly indicate parts that are not to be graded.
- Duration: 75 minutes.

Problem 1: Raster Graphics Pipeline.

(4+3=7 points)

The following steps are part of the raster graphics pipeline: back-face culling, depth buffering, scan conversion, projection, anti-aliasing, transformations, and view frustum culling.

- (a) Bring the steps in the right order. Briefly explain your answer.
- (b) Ray casting is an alternative to the above pipeline. Which steps are obsolete when performing a ray casting and which can be combined with ray casting? Briefly explain your answer.

Problem 2: Projection.

(10+5=15 points)

Given a triangle Δ with vertices $\mathbf{v}_1 = (0, 0, 3)$, $\mathbf{v}_2 = (1, 4, 3)$, and $\mathbf{v}_3 = (3, 1, 3)$ in Cartesian coordinates. In the same coordinate system, we define a view point $\mathbf{p} = (0, 0, -2)$ and the 2D screen coordinates by its origin $\mathbf{o} = (0, 0, 0)$, and the two axes $\mathbf{e}_1 = (1, 0, 0)$ and $\mathbf{e}_2 = (0, 1, 0)$.

- (a) Starting with the geometric constellation, derive the perspective projection matrix in homogeneous coordinates for the scene described above.
- (b) Apply the homogeneous perspective projection matrix to the given triangle and report the projected triangle.

Problem 3: Ray Casting and Anti-aliasing.

(10+4=14 points)

Given the same constellation as in Problem 2, i.e., triangle Δ with vertices $\mathbf{v}_1 = (0, 0, 3)$, $\mathbf{v}_2 = (1, 4, 3)$, and $\mathbf{v}_3 = (3, 1, 3)$, and view point $\mathbf{p} = (0, 0, -2)$. Consider one(!) pixel of the screen, namely, the one given by the quadrilateral $\mathbf{v}_1 = (0, 0, 0)$, $\mathbf{v}_2 = (1, 0, 0)$, $\mathbf{v}_3 = (1, 1, 0)$, and $\mathbf{v}_4 = (0, 1, 0)$.

- (a) Apply ray casting to the given pixel, i.e.:
- Compute the intersection of the respective ray with the plane of the triangle and report back the intersection point.
 - Compute the barycentric coordinates of the intersection point with respect to the triangle and report whether the intersection point lies within the triangle.
- (b) What is aliasing, what is anti-aliasing, and how can we efficiently compute anti-aliasing within the ray casting framework? Note that you should explain how the anti-aliasing can be computed, but you do not need to compute it for the given example.

Problem 4: Scan Conversion.

(10+5=15 points)

Given a triangle Δ' with projected vertices $\mathbf{v}_1 = (\frac{5}{2}, \frac{1}{2})$, $\mathbf{v}_2 = (\frac{15}{2}, \frac{3}{2})$, and $\mathbf{v}_3 = (\frac{9}{2}, \frac{11}{2})$ in the screen-space's Cartesian coordinates. Assume that the origin of the screen is at $(0, 0)$, that the pixels are of size 1×1 , and that we have a screen of size 9×6 pixels (i.e., the screen stretches over the interval $[0, 9] \times [0, 6]$).

- (a) Apply the Bresenham algorithm to scan convert the edges of the triangle. Provide all computation steps including the error values after each iteration step.
- (b) Starting with the barycenter of the triangle, apply a seed-fill algorithm to scan convert the triangle. Explain your computation steps.

Problem 5: Depth Buffering and Color Computation.*(10+4=14 points)*

Given the same constellation as is Problem 4, i.e., we assume triangle Δ' with projected vertices $\mathbf{v}_1 = (\frac{5}{2}, \frac{1}{2})$, $\mathbf{v}_2 = (\frac{15}{2}, \frac{3}{2})$, and $\mathbf{v}_3 = (\frac{9}{2}, \frac{11}{2})$ in the screen-space's Cartesian coordinates, and we assume that the origin of the screen is at $(0, 0)$, that the pixels are of size 1×1 , and that we have a screen of size 9×6 pixels (i.e., the screen stretches over the interval $[0, 9] \times [0, 6]$). Moreover, we assume that the depth of the triangle is constant of value 5.7 and that its color is yellow with 50% transparency. Before processing the triangle, the depth buffer contains depths $i + j$ for each pixel (i, j) and the framebuffer contains constantly a blue color.

- (a) Process the triangle by updating framebuffer and depth buffer. Explain your steps. Note that you shall not consider any antialiasing. Also note that you may assume an approximate scan conversion result such that you can complete this question even without having solved Problem 4.
- (b) Convert the colors from RGB to CMYK color format.