Final Exam: Monday 2:50–5:40

1 Rules

Closed book, you can use your notes (3 pages total, must be handwritten). All the obvious rules apply, of course.

2 Coverage

Only things described in the class notes posted on the class web page (and the book sections they refer to, especially *shading models* material) will be covered. As always, the more you know the better, but here is what you can safely concentrate on.

Simple ray tracing: 3D vectors, intersections, local reflection models (know how to compute ray-sphere and ray-triangle intersections, decide visibility, and shadows and do shading calculations (normals, reflected directions, diffuse and specular term etc))

3D pipeline: Transformations, homogenous coordinates, shading models (flat, Gouraud, Phong), back-face culling, z-buffer, linear interpolation, clipping, triangle strips and fans, Bresenham algorithm (lines and circles), triangle and polygon scan-conversion. (sample problems: building transformations from the simple ones by composing/multiplying matrices, comparison of shading models, Given two triangles, how to tell that one is front facing and one is back facing? Build strips and fans for a simple triangle mesh. Explain the meaning and use of the decision variable in Bresenham-type algorithms; Why do people use homogenous coordinates?! Give the most important reasons)

3D Modeling and Data Structures: Half-edge representation, scene graphs, Bezier curves, B-splines, subdivision for curves and surfaces, BSP trees (Scene graphs to opengl conversion; Joining Bezier curves smoothly; how to build BSP and shadow BSP trees? Painter's algorithm; Half-edge data structure operations and algorithms (listing incident triangles for a vertex and such))

Texturing: Texture coordinates, shadows on a plane, mirror reflections, bilboards, environment mapping (understand the texturing process: how is color looked up for a triangle with given texture coordinates at the vertices? Know and understand the tricks: shadows on a plane, bilboards, mirrors, environment mapping)

Rendering Equation: Be sure to understand BRDF and the overall idea of the rendering equation (and can explain it yourself... I can imagine a problem where I write the rendering equation in a slightly different form, maybe using a different notation and ask what things mean and why they are where they are)

Shadow volumes: Understand the algorithms (sample question: a 'zfail'-type shadow volume algorithm works exactly like what the one we covered in class ('zpass'), but, for each pixel, counts intersections with shadow polygons that

happen *behind* the closest intersection point, not in front of it. Explain why this algorithm works and detail its main stages).

Quaternions: know how to operate on them and how to use them to represent rotations (Example question: Explain how to represent a rotation by 90 degrees around the x-axis using quaternions. Then, represent this rotation as a matrix and show that the two representations are equivalent, i.e. rotating a vector with quaternions and the matrix gives the same result)