CMSC427 Final review fall 2017

Questions will come from homeworks 1 to 6, the quiz and midterm, the labs and the projects. Any additional questions will be listed here, or on a practice final given out on Thursday. The final will be about 10 pages.

The final is 10:30-12:30 on Tuesday, Dec. 19th. I will be in Friday the 15th before 4 pm (my 250 final is 4-6), and on Monday the 18th for the afternoon. Also Thursday the 14th before 3:30, but not in on Wednesday the 13th.

How to run OpenGL or GLSL will not be on the exam, although the basics of where OpenGL client programs, and shaders, run is fair game.

Topics and questions which are likely to be on exam (somewhat random in order):

1) Convert between point-point, point-vector, point-normal forms of line (see relevant handout).
2) Compute dot and cross products, normalize a vector (magnitude)
3) Compute distance between line and point, project vector onto two others.
4) Compute camera matrix for general or special cases.
5) Explain the role of at, lookAt, up.
6) Explain and demonstrate how to turn, rotate camera
7) Blend two parametric curves into surface patch.
8) Compute normal vector for a parametric patch.
9) Explain what CSG is and what is for.
10) Explain how the parametric function x(u,v) draws a parametric patch
11) Write out an equation for the function lerp(u,p1,p2)
12) Develop an equation for a bilinear patch from scratch
13) Explain the process of subdivision surfaces, eg show how you might subdivide a triangle into four triangles.
14) Given a set of points, sketch a curve that satisfies location and tangent constraints
15) Explain and apply C0, C1 and C2 continuity
16) Given a set of points and contraints, compute the parametric coefficients for linear, quadratic, or cubic curve.
18) Given a set of points, create a geometry vector (eg, from points get tangents)
19) Given a graph of blending curves for a type of parametric curve
20) Explain the improvement of Catmull-Rom curves over Hermite curves, and show the construction for Catmull-Rom.
21) Given a parametric curve as a polynomial, rewrite it to show the blending functions.
22) Compute midpoint, perpendicular bisector – compute perp vector.
23) Use linear interpolation equation between two points (ie, tweening).
24) Compute reflection vector.
25) Convert between 3 point, point-two vector, point-normal form of plane
26) Know turtle graphics and relative moves.
27) Show that implicit equations satisfy a parametric equation.
28) Show how to draw a parametric equation in OpenGL.

29) Be able to define and have an idea on how to compute polygon properties like simplicity, convexity, and winding.
30) Develop a surface of rotation from parametric curve.
31) Show the steps of the de Casteljau algorithm for Bezier curves.
32) Compute tangent of a parametric curve, and the (approximate) Frenet frame.
33) Know the difference between an interpolating and approximating curve.
34) Know the properties of a Bezier curve (convex hull, variation diminishing, affine invariance).
35) Work with the vector/matrix form for a parametric curve.
36) Describe the process of 3D capture of a 3D mesh.
37) Define the shadow terms umbra and penumbra.
38) Describe the two pass algorithm for shadow mapping.
39) Define point, directional, area and spot lights.
40) Describe and apply Z-buffering for depth, and explain why numeric resolution is an issue.
41) Explain environment maps and reflection mapping, and how it can be handled by cube maps.
42) Explain what ambient occlusion.
43) Explain toon shading.
44) Work with equation for refraction.
45) Explain difference between ray tracing and radiosity for global illumination.
46) What's the Cornell box?
47) Explain the set of rays used in ray tracing.
48) Explain how texture mapping works for basic surfaces, including parametric.
49) Explain how texture map data can be used for color, for setting specularity, for setting any surface property.
50) Explain anti-aliasing in texture mapping, and how mipmaps are used to combat that.
51) Use recursive rule rewriting (with generator) to create fractal curve.
52) Compute fractal dimension of a fractal curve given a generator (on a grid).
53) Draw the shape represented by Turtle command string.
54) Use an L-system grammar to generate stages (generations) of a curve.
55) Define evolutionary art and particle system.
56) That enough?

A few topics discussed or on slides, but *not* on the exam.
A) Ruled surfaces – nice topic, not hard to know basic idea, but no question.
B) Berstein polynomials for Bezier curves
C) Adaptive sampling recursive subdivision for parametric curve.
D) Shadow volumes.
E) Fresnel equations.
F) Ray marching, path tracing.
G) Bump mapping.
H) Mandelbrot sets