

CISC 440/640

Final

Thursday, December 18, 2008

Only graduate students are required to answer questions 2 and 4 of this exam. Point totals will be normalized to a common scale (the exam is worth 25% of your grade). Partial credit will be given in quarter-point gradations.

1. (4 points)

Suppose we want to write a ray tracing function that checks whether a ray $\mathbf{p} = \mathbf{o} + t \mathbf{d}$ intersects a general polygon consisting of co-planar vertices $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$. Describe in detail how to do this.

2. (Graduate students only -- 5 points)

Using the regular expression representation of light paths discussed in class, what kinds of paths can basic ray tracing handle? What kinds of paths does *bidirectional ray tracing* allow? Explain the key steps and data structures used in bidirectional ray tracing in as much detail as you can.

3. (3 points)

Define *magnification* and *minification* in the context of texture mapping. Why, specifically, does bilinear interpolation not always yield good results for the latter? What OpenGL function is used to select the texture filtering method for magnification and minification?

4. (Graduate students only -- 3 points)

If we have computed the texture coordinates (s, t) for each vertex of a triangle in 3-D, explain what is wrong with simply linearly interpolating them to the triangle interior during rasterization.

5. (4 points)

Define *environment maps* and discuss the pros and cons of using them vs. ray-tracing.

6. (3 points)

Consider a quadratic Bezier curve with control points $(0, 0)$, $(1, 1)$, and $(2, 0)$. What are the coordinates of the curve at $t = 0.3$? Show your work.

7. (2 points)

Define C^0 , C^1 , and C^2 continuity of curves. If we define a piecewise quadratic Bezier curve such that there are three control points per segment and the last control point for each segment is used as the first one for the next segment, what kind of continuity would this curve have?

8. (3 points)

Explain the difference between approximating and interpolating splines and name one specific type of each (excluding Bezier). What is a graphics application that interpolating spline curves might be more appropriate for than approximating spline curves, and vice versa?

9. (4 points)

Suppose we want to animate a *flag* (outdoors, on a flagpole) with a particle system. What would the particles mean here and what are all of the forces that might need to be simulated for a realistic result?

10. (2 points)

What are the three basic “behaviors” in Reynolds' original flocking algorithm? What other behaviors might be added for more control or realism?