

## CPSC 352 Final Exam Information

- Date: **Wednesday, May 17, 9am or** \_\_\_\_\_
- Exam is **cumulative**. There will be more emphasis on the last 1/3.
- Format: Similar to tests 1 and 2. Short answer (e.g. “Explain the Gouraud Shading algorithm”); some more concrete problems (e.g. show how to find the angle between two vectors [given] using the dot product, show homogeneous transformation matrices for scaling around the point [1, 2, 3], give the order in which the faces in the shown BSP tree would be displayed when it is traversed from the given viewpoint)
- Use the review sheets from tests 1 and 2 (linked on the syllabus web page)

### Additional topics since test 2

#### Chapter 8: The Rendering Pipeline

1. Front-end vs. back-end processing: what is done in each?
2. Coordinate systems: object, eye, clip, NDC, screen
3. Clipping
  - a. How to compute the intersection of a line  $(x_1, y_1)-(x_2, y_2)$  with a line, e.g.  $y=y_1$
  - b. Cohen-Sutherland clipping algorithm (skip Liang-Barsky, Sutherland-Hodgeman)
  - c. Trivial accept-reject
4. Scissoring
5. Rasterizing a line
  - a. DDA Algorithm
  - b. Bresenham’s algorithm (conceptually; you don’t have to memorize equations)
6. Rasterizing polygons
  - a. Winding number, inside-outside rules
  - b. Flood fill
  - c. Scan-line approach
  - d. Handling singularities (e.g. a scanline passing through a vertex)
7. Anti-aliasing, temporal anti-aliasing
8. Color systems: RGB, CMYK, YIQ, HLS
9. Halftoning
10. Dithering
  - a. Patterned dither
  - b. Floyd-Steinberg error-diffusion dither algorithm

#### Chapter 9: Hierarchical graphics, modeling, and animation

1. Primitives such as points, lines, polygons, ellipses, NURBS, particles, meshes, skin and bones, subdivision surfaces, algorithmic primitives
2. Meshes and mesh deformations: e.g. modeling landscapes, cloth
3. Hierarchical models, parenting
4. Animation with skin and bones
5. Kinematics; inverse kinematics
6. Rigging
7. Keyframe animation
8. Motion capture, morphing
9. Tree-based data structures for representing objects
  - a. Quadtree, Octree
  - b. BSP-tree and its use for hidden surface removal
  - c. Algorithms for building, traversing BSP-trees

## Sample Questions

1. What is texture mapping? Why do we have to give texture coordinates when using texture mapping? Explain the methods that WebGL uses to deal with aliasing.
2. Explain how billboarding might be used to render a tree with only two polygons. How does this technique use the stencil buffer?
3. Explain normal mapping.
4. Why are hierarchical DAG models used to represent complex objects? What is the purpose of the transformation associated with each edge of the DAG? What is the algorithm for rendering such models?
5. What is the difference between kinematics and inverse kinematics? How would inverse kinematics be combined with keyframing for animating the model?
6. What is motion capture? How might it be combined with inverse kinematics and/or biomedical simulation to make realistic character animation
7. How could you use keyframing to animate a 3D morph of an object defined with curved surfaces such as NURBS?
8. Explain the BSP-tree insertion and traversal algorithms. How is it used for hidden surface removal in the absence of a depth buffer?
9. Given a line segment from  $(x_1, y_1)$  to  $(x_2, y_2)$ , how would you compute the intersection parameter with the  $y=y_0$  line?
10. What are the Cohen-Sutherland “outcodes” and how are they used in their clipping algorithm?
11. What is “scissoring” when would you want to (or not want to) use it?
12. Explain the Floyd-Steinberg error-diffusion dither algorithm.