ECE 478 Computer Graphics

Final Examination

26th April 2011

2 hours

Name:

Student Number:

Marks will be given for demonstrating your thought processes as well as the correct answer. **Make sure you show all the steps you use for each of your answers.** If you do not know the answer to a question, write down the process by which you think it could be answered.

Question 1

a) Explain the concept of outcodes in the Cohen-Sutherland clipping algorithm for a 2D viewport. **(2 marks)**

b) Using line segments for your examples, draw four different cases of clipping a line against a 2D viewport. **(4 marks)**

c) How would you extend this approach to work as a pipeline? **(1 mark)**

d) How would you extend the Cohen-Sutherland algorithm to 3D clipping (i.e. clipping against a volume instead of a window)? **(1 mark)**

Question 2

a) Why are blending and translucency important in computer graphics? **(1 mark)**

b) Why is the order of rendering important when blending? **(1 marks)**

c) Which buffer needs to be treated differently when rendering scenes with translucency? **(1 mark)**

d) Why is blending required for most anti-aliasing techniques? **(1 mark)**

e) Describe one approach for anti-aliasing line segments. **(2 marks)**

Question 3
A company wants to create an architectural renderer for a world which contains many cities, each city contains many buildings and each building is made up of two components.

a) Give two advantages of using hierarchical modelling or scene graphs to model a scene. **(2 marks)**

b) Draw an example tree structure (basic hierarchical model) to model a world with two cities, each city has two buildings and each building has two components. Label each node in your tree with a letter (e.g. node **A**). Label each *edge* connecting two nodes together as $\mathbf{M_{AB}}$ to represent the transformation from node **A** to node **B** (you do not need to specify what the transformations are). **(4 marks)**

c) For each component at the bottom of the tree (i.e. the leaf nodes), write out the series of transformations required to transform this component (assuming the world has already been initialized). **(2 marks)**

d) Assuming an OpenGL-like system: for one component only, write pseudo-code to draw the component with the correct transformation. Use the functions **load_identity()**, **multiply($\mathbf{M_{AB}}$)** and **draw_node(N)** for your pseudo-code. (Ignore the matrix stack for this question, just load the full transformation.) **(2 marks)**

Question 4

a) What are the two main types of shader used in programmable pipelines, and which component provides the input to each? **(4 marks)**

b) Give one advantage of using programmable pipelines instead of fixed-function pipelines. **(1 mark)**

c) Briefly describe the inputs and outputs of the two types of shaders. **(2 marks)**

Question 5

a) Assuming an OpenGL-like system, describe the coordinate system used to index textures (provide the origin, axes and values). **(2 marks)**

b) Describe the approaches of point sampling and linear filtering for applying textures. **(2 marks)**

c) Pixels and texels rarely map one-to-one: for this case, which two operations must be defined for textures to render correctly? **(2 marks)**

d) In which case is mip-mapping applied? How is a mip-map constructed? **(2 marks)**

Question 6

a) Is it possible to construct a translation matrix for 3D objects using a 3x3 matrix? Give an explanation for your answer. **(2 marks)**

b) Why can the modelview matrix not be used directly to transform surface normals? (Hint: a diagram may help) **(2 marks)**

c) Given a polygon made up of a set of more than three 3D vertices, devise a test to determine if the polygon they represent is planar. (Provide an example of your test.) **(3 marks)**

d) Find the projection of a point onto the plane *ax + by + cz + d = 0* from a light source located at infinity in the direction (*dx, dy, dz)*. **(4 marks)**