DUBLIN CITY UNIVERSITY

SEMESTER TWO EXAMINATIONS 2009

MODULE:	Graphics and Visualisation EE563
COURSE:	MTC – M.Eng. in Telecommunications Engineering MEN – M.Eng. in Electronic Systems GDE – Graduate Diploma in Electronic Systems
YEAR:	Postgraduate (C)
EXAMINERS:	Dr Robert Sadleir, Section A (x8592) Dr Derek Molloy, Section B (x5355) Prof. Paul Rees
TIME ALLOWED:	3 Hours
INSTRUCTIONS:	Please answer any TWO questions from Section A and any TWO questions from Section B. All questions carry equal marks.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO

The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones

Section A

Question 1

(a) Answer all of the following short questions (make sure to keep your answers concise):

(i) Describe the type of data that is represented by the following classes from the `javax.vecmath` package: `Color4f` and `Point2d`.

(ii) What are capability bits used for in Java3D? Give an example of a pair of read/write capabilities that are associated with a particular Java3D node or node component. What are the default settings for these capabilities?

(iii) What is the purpose of each of the following vertex format flags: `COORDINATES`, `NORMALS` and `TEXTURE_COORDINATES_2`? Describe how all three of these flags can be passed simultaneously to the constructor of one of the subclasses of `GeometryArray`.

(iv) What is the benefit associated with using strip geometry when defining continuous surfaces. Give an example that demonstrates how one type of strip geometry works.

(v) Describe the operation of the three face culling modes supported by the `PolygonAttributes` appearance component. Which of these is the default mode?

(vi) Compare and contrast the coordinate systems used by Java 2D images and Java textures.

(vii) What are the `PositionInterpolator` and `Billboard` behaviours used for?

[14 marks]

(b) Determine the coefficients of the transformation matrices that represent the following 2D transformations:

(i) A scaling of 50% in the horizontal direction and 200% in the vertical direction.
(ii) A translation of 7 pixels in the horizontal direction and 3 pixels in the vertical direction.

What is the result of applying these transformations sequentially to the line connecting (-3, 8) to (4, 2)? **Note:** The transformation from one 2D coordinate system to another is represented by the following equation:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

[6 marks]

(c) Describe how the effect of mapping a different texture to each face of a VRML Cylinder primitive can be achieved.

[5 marks]

Question 2

(a) Describe in detail the scene graph structure used by Java3D (i.e. the *directed acyclic graph*). Your description should make particular reference to group nodes, leaf nodes, node components and the types of relationships that can exist between these scene graph elements.

[9 marks]

(b) Provide initial values for the arrays of points and indices from Listing 2 (b) in order to generate the prism shape illustrated in Figure 2 (b). **Note:** All exterior faces of the prism are front facing. How much additional memory would be required to define this structure if a **`TriangleArray`** was used instead of an **`IndexedTriangleArray`** given that Java **`int`** and **`float`** primitives both require 4 bytes of storage?

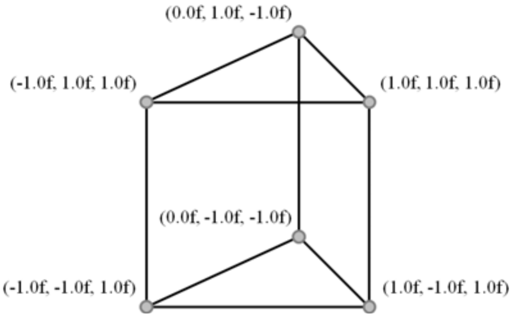

Figure 2 (b): A prism shape and its vertex coordinates.

```
public BranchGroup createContentBranch()
{
  BranchGroup root = new BranchGroup();

  float[] points = { ___, ___, ___,
                     ___, ___, ___,
                     ___, ___, ___,
                     ___, ___, ___,
                     ___, ___, ___,
                     ___, ___, ___ };

  int[] indices = { _, _, _,
                    _, _, _,
                    _, _, _,
                    _, _, _,
                    _, _, _,
                    _, _, _,
                    _, _, _,
                    _, _, _ };

  IndexedTriangleArray geometry = new IndexedTriangleArray(8,
    GeometryArray.COORDINATES, 24);
  geometry.setCoordinates(0, points);
  geometry.setCoordinateIndices(0, indices);

  Shape3D prism = new Shape3D(geometry);

  root.addChild(prism);
  root.compile();
  return root;
}
```

Listing 2 (b): Skeleton Java3D code for the prism shape illustrated in Figure 2 (b).

[6 marks]

Question 2 continued overleaf…

(c) Describe how a **`TexCoordGeneration`** object generates texture coordinates using the **`OBJECT_LINEAR`** texture generation mode. What texture coordinates would be generated for the vertices of the prism in Figure 2 (b) using this mode.

[4 marks]

(d) Briefly describe the purpose of the following four appearance components:

(i) **`ColoringAttributes`**
(ii) **`Material`**
(iii) **`LineAttributes`**
(iv) **`TransparencyAttributes`**

[6 marks]

Question 3

(a) What is the purpose of the Java3D **`DistanceLOD`** behaviour? Describe, with the aid of a scene graph diagram, how this type of behaviour operates in conjunction with a **`Switch`** node. Assume that the **`Switch`** node has three children as illustrated in Figure 3 (a) and that the **`DistanceLOD`** was constructed using the float array {5.0f, 25.0f}.

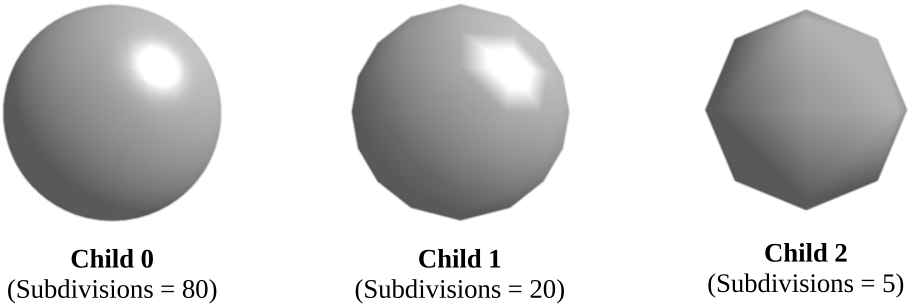

Figure 3 (a): A series of Sphere primitives with different numbers of subdivisions.

[9 marks]

(b) For each of the voxel configurations illustrated in Figure 3 (b) indicate **two** possible adjoining configurations that would result in a single continuous surface being generated. In each case, provide illustrations of the combined configurations and their associated surface patches. The standard marching cubes algorithm sometimes fails to generate continuous surfaces where it would be appropriate to do so. Explain why this is the case and describe how the standard algorithm can be updated to solve this problem.

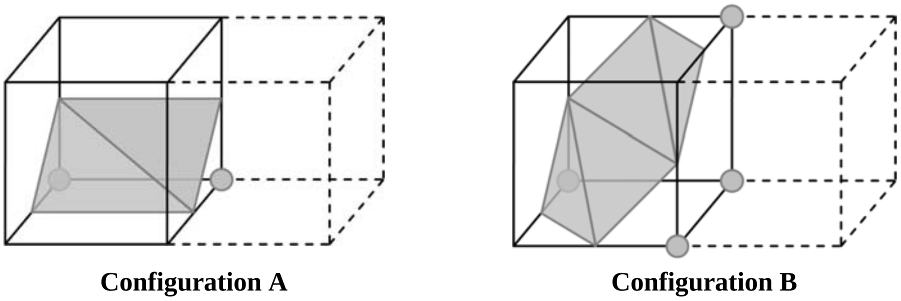

Figure 3 (b): Two voxel configurations (A & B) with unspecified adjoining configurations (illustrated using dashed lines)

[8 marks]

(c) What is meant by the term *direct volume rendering*? Describe two different approaches to direct volume rendering and use illustrations to support your descriptions.

[8 marks]

Section B

Question 4

(a) If you were assigned the task of designing a ‘first person shooter’ computer game where the screen represents the view (through the eyes) of an army solider who is in a combat scenario, describe all the different C++ classes that you would design and how they would relate together in a scene graph. What other computer graphics techniques would you employ to make the game run efficiently (run fast)?

[8 marks]

(b) Using C++ pseudo-code, write a generic container class for your scene graph objects described in (a) that is capable of storing and identifying scene graph elements.

[9 marks]

(c) Using C++ pseudo-code, write an algorithm for traversing your scene graph tree containers from part (b) in a recursive manner. List the various reasons that you would traverse the scene graph, given the gaming application as described in part (a).

[8 marks]

Question 5

(a) What is GLUT? And, why would you use it with OpenGL?

[4 marks]

(b) Describe accurately what is displayed after a call to the function **`draw()`** in Listing 5 (b) below. Explain step by step what is happening in these functions.

```
void draw()
{
  glColor(1,0,0);
  glPushMatrix();
  glScale(3,3,3);
  glTranslate(2,0,0);
  glRotate(90, 0,0,1);
  glPushMatrix();
  glTranslate(1,0,0);
  drawSquare();
  glColor(0,1,0);
  glPopMatrix();
  glPushMatrix();
  glTranslate(4,0,0);
  glScale(0.5,0.5,0.5);
  glRotate(270, 0,0,1);
  drawSquare();
}

void drawSquare()
{
  glBegin(GL_QUADS);
  glVertex(0,0,0);
  glVertex(0,1,0);
  glVertex(1,1,0);
  glVertex(1,0,0);
  glEnd();
}
```

Listing 5 (b): Open GL source code.

[8 marks]

Question 5 continued overleaf…

(c) Describe the OpenGL retained mode and immediate mode with reference to the OpenGL client/server model. Discuss the advantages and disadvantages of each mode. Use a short segment of pseudo-code to outline how you would define and use a display list in OpenGL. How would you adapt the code in Listing 5 (b) to use retained mode graphics?

[8 marks]

(d) Describe the concepts clipping and culling. Explain why they are used and describe a simple test for both conditions on a particular polygon.

[5 marks]

Question 6

(a) Give an example where the order that objects are drawn matters when using a z-Buffer.

[4 marks]

(b) What is the difference between Bump Mapping and Displacement Mapping? Describe a situation where you can tell the difference. What vertex-information must exist in order to apply bump-mapping? Why is this information needed? Explain how it is used in the algorithm for bump-mapping.

[7 marks]

(c) Describe the difference between shading and shadowing, and explain the concepts of umbra and penumbra. Describe in detail any **<u>two</u>** real-time approaches/algorithms for computing shadows. Relate the real-time approaches to how shadows are computed in ray-casting.

[14 marks]