DUBLIN CITY UNIVERSITY

SAMPLE EXAMINATION 2007

MODULE: ***(Title & Code)***	3D Graphics and Visualisation EE563
COURSE:	M.Eng./Grad. Dip./Grad. Cert. in Electronic Systems M.Eng./Grad. Dip./Grad. Cert. in Telecommunications Engineering ECSAX/GDE/GDEI/MEN/MENI/MTC/MTCN
YEAR:	Postgraduate (C)
EXAMINERS:	Dr Robert Sadleir (x8592) Dr Derek Molloy (x5355)
TIME ALLOWED:	3 Hours
INSTRUCTIONS:	Please answer FOUR questions. All questions carry equal marks

Requirements for this paper
Please tick (X) as appropriate

- ☐ ***Log Table***
- ☐ ***Graph Paper***
- ☐ ***Attached Answer Sheet***
- ☐ ***Statistical Tables***
- ☐ ***Floppy Disk***
- ☐ ***Actuarial Tables***

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.

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

Question 1

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

(i) What format does a **`Color4f`** object use to store colour information? How does this representation of colour information differ from the approach used in the default RGB colour model where a colour is represented in the form: **`0xAARRGGBB`**?

(ii) List the five colour properties defined by the **`Material`** class. Which one of these colour properties is unaffected by illumination?

(iii) What is the purpose of a **`SharedGroup`** node? Describe the relationship between a **`SharedGroup`** node and a **`Link`** node in a scene graph. Provide a sample scene graph to support your answer.

(iv) If an ambient light with a purple colour (0.45, 0.20, 0.93) illuminates an object with a red ambient colour (0.78, 0.15, 0.23) what colour will the object appear to be in the rendered scene?

(v) What is the relationship between a **`Shape3D`** node and an **`Appearance`** node? Give examples of two node components that an **`Appearance`** node might maintain references to.

(vi) There are two texture boundary modes defined by the **`Texture`** class: **`CLAMP`** and **`WRAP`**. Briefly describe the effect of these two boundary modes.

(vii) What is picking and how does it relate to viewing. Picking is supported by Java 3D using the **`PickCanvas`** class. What is the purpose of the **`setTolerance()`** method of the **`PickCanvas`** class?

[14 marks]

(b) A line is defined by its two endpoints. Using the equation below scale the 2D line connecting (-5, -8) to (3, 3) by 50% in the horizontal direction and 200% in the vertical direction.

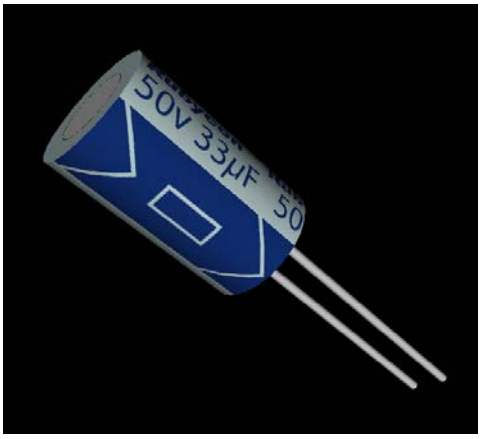
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

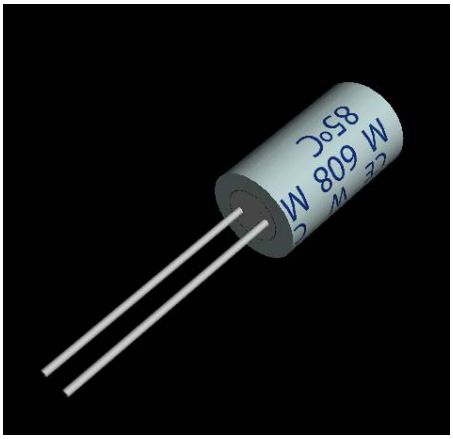
Briefly describe two other transformations that can be applied to a 2D image.

[5 marks]

Question 1 continued overleaf…

(c) Describe in detail how the capacitor model illustrated below could be generated using VRML. Note that the top and bottom of the main body of the capacitor have the same appearance.





[6 marks]

Question 2

(a) Briefly describe the functionality of the following subclasses of the **`Behavior`** class:

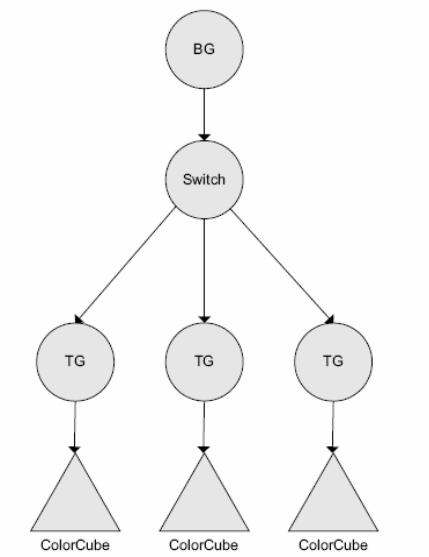
(i) **`DistanceLOD`**
(ii) **`MouseZoom`**
(iii) **`Billboard`**
(iv) **`PositionInterpolator`**

[8 marks]

(b) What are the differences between **`TriangleArray`**, **`TriangleStripArray`** and **`TriangleFanArray`** geometries? Give an example of each of these types of geometry to illustrate your answer.

[6 marks]

(c) Describe the 3D scene represented by the following **`BranchGroup`** rooted scene graph. Explain how the **`Switch`** node can be used to control how its children are rendered.



[5 marks]

Question 2 continued overleaf…

(d) Explain, in detail, the operation of the following code and draw an illustration of the expected outcome when the code is executed.

```
 1.  public BranchGroup createContentBranch()
 2.  {
 3.    BranchGroup root = new BranchGroup();
 4.
 5.    float[] points = {-0.5f, -0.5f, -0.5f,
 6.                       0.5f, -0.5f, -0.5f,
 7.                      -0.5f,  0.5f, -0.5f,
 8.                      -0.5f, -0.5f,  0.5f,
 9.                      -0.5f,  0.5f,  0.5f,
10.                       0.5f, -0.5f,  0.5f,
11.                       0.5f,  0.5f, -0.5f,
12.                       0.5f,  0.5f,  0.5f};
13.
14.    int[] indices = {0, 3, 4, 2,
15.                     0, 1, 5, 3,
16.                     0, 2, 6, 1,
17.                     7, 5, 1, 6,
18.                     7, 6, 2, 4,
19.                     7, 4, 3, 5};
20.
21.    IndexedQuadArray quadArray = new IndexedQuadArray(8,
22.      GeometryArray.COORDINATES, 24);
23.    quadArray.setCoordinates(0, points);
24.    quadArray.setCoordinateIndices(0, indices);
25.
26.    Shape3D cube = new Shape3D(quadArray);
27.
28.    root.addChild(cube);
29.    root.compile();
30.
31.    return root;
32.  }
```

[8 marks]

Question 3

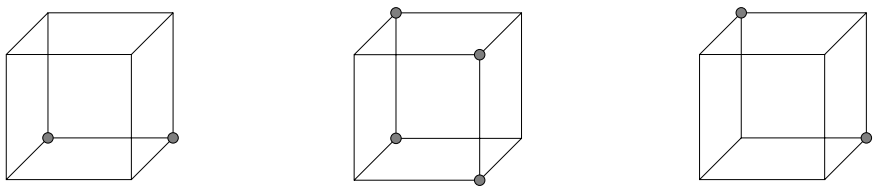
(a) What are B-Splines? Describe how B-Splines are generated with reference to the following formula:

$$Q_i(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}$$

Why are B-Splines useful for generating curved paths in 3D space?

[9 marks]

(b) Describe the operation of the marching cubes algorithm. Draw the surface patches that correspond to the following voxel configurations according to the marching cubes algorithm. Note that a corner sphere indicates the presence of a voxel inside the isosurface, whereas the omission of a corner sphere indicates the presence of a voxel outside the isosurface.



Discuss one of the problems associated with the standard marching cubes algorithm and suggest a possible solution to this problem.

[8 marks]

Question 3 continued overleaf…

(c) Describe the meaning of the following volume rendering terms:

(i) Maximum intensity projection
(ii) Voxel
(iii) Splatting
(iv) Compositing

[8 marks]

Question 4

(a) Explain the following concepts: clipping, culling, and frustum. How do they relate?

[3 marks]

(b) Describe the OpenGL Current Transformation Matrix (CTM). Describe two different ways that would allow you to rotate an object about an arbitrary point *P(x,y,z)* when using OpenGL. Describe the OpenGL calls that you would make.

[7 marks]

(c) When designing a C++ class for representing vectors or matrices there are many design decisions related to the C++ language. Describe how you would use the following concepts in your design: the inline keyword; operator overloading; data types; passing by const reference; using const methods; using virtual methods, and the use of encapsulation.

[7 marks]

(d) Draw and describe the output of this segment of C++ OpenGL code. Please also describe the individual OpenGL calls that are used in this segment:

```
00  // Called to update the scene
01  int drawGLScene(float theta)
02  {
03      glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
04      glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
05
06      glMatrixMode(GL_PROJECTION);
07      glLoadIdentity();
08      gluPerspective(60.0, 1.0, 1.0, 20.0);
09
10      glMatrixMode(GL_MODELVIEW);
11      glLoadIdentity();
12      gluLookAt(0.0, 0.0, 3.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
13      glRotatef (theta, 1.0f, 1.0f, 0.0f);
14
15      glPushMatrix();
16      glColor3f(0.0f, 1.0f, 0.0f);
17      glCallList(SPHERE);
18      glPopMatrix();
19
20      glPushMatrix();
21      glColor3f(0.0f, 0.0f, 1.0f);
22      glTranslatef(1.0f, 0.0f, 0.0f);
23      glCallList(SPHERE);
24      glPopMatrix();
25
26      glPushMatrix();
27      glColor3f(1.0f, 0.0f, 0.0f);
28      glTranslatef(-1.0f, 0.0f, 0.0f);
29      glCallList(SPHERE);
30      glPopMatrix();
31      return TRUE;
32  }
```

[8 marks]

Question 5

(a) Describe the Constructive Solid Geometry (CSG) approach to representing complex geometries. Give an example of a CSG tree and an associated object.

[6 marks]

(b) Outline the general algorithm used in building BSP Trees.

[3 marks]

(c) Apply the algorithm outlined in (b) to step-by-step subdivide the scene as illustrated in Figure 4.1(a) and to build a representative BSP Tree.

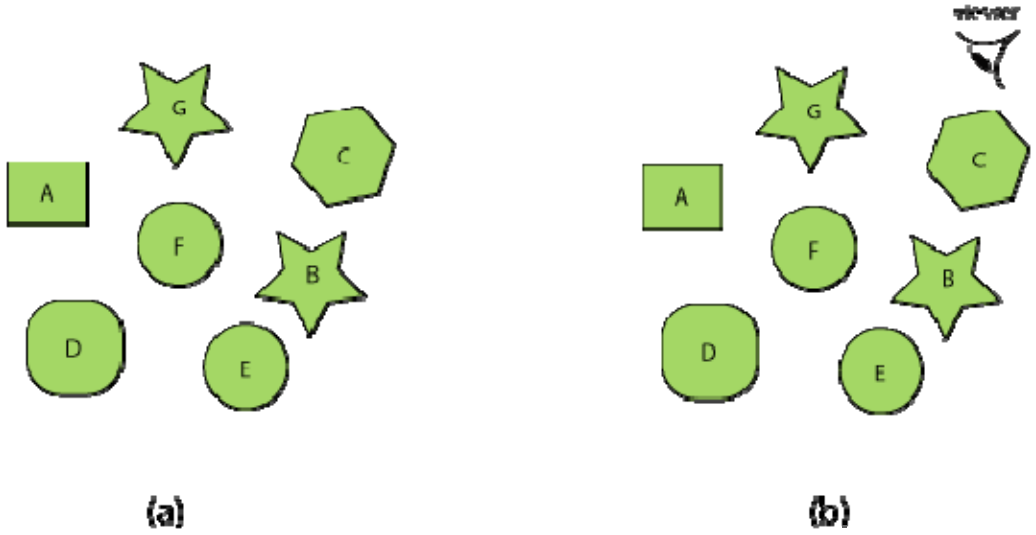


Figure 4.1 (a) Example scene with objects A to G (b) illustration of a viewer in the top right-hand corner, facing towards the centre of the object cluster (object F).

[6 marks]

(d) Describe the Painter's Algorithm. Why do computer graphic applications reverse this algorithm? Describe an example scene where the Painter's algorithm will fail.

[4 marks]

(e) Why are BSP Trees useful when building hidden surface removal algorithms? Using the BSP Tree that you built in (c) illustrate how you could perform a search to find the visibility order of all of the objects in the scene if the observer was placed in the top right-hand corner, facing towards the centre of the object cluster (as illustrated in Figure 4.1(b)).

[6 marks]

Question 6

(a) Describe lighting in OpenGL. Describe the four light types and how they differ. Describe the different properties that light sources can have. Why do lights have a light number? Describe the use of the following OpenGL function:

```
glLightfv(GLenum source, GLenum parameter, GLfloat _pointer to array);
```

[8 marks]

(b) Using C++ pseudo-code, write a generic container class for a Scene Graph Tree that is capable of storing and identifying scene graph elements, such as lights, objects, cameras etc.

[12 marks]

(c) Using C++ pseudo-code, write an algorithm for traversing your scene graph tree containers from part (c) in a recursive manner.

[5 marks]