DUBLIN CITY UNIVERSITY

SEMESTER TWO EXAMINATIONS 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:	Prof. Paul Rees 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 two node component objects does a **`Shape3D`** object maintain references to? Explain what the referenced node components represent.

(ii) A sphere has a radius of 1 metre and colouring attributes with a red colour (1.0, 0.0, 0.0). The sphere is in the presence of linear fog with a back distance of 25 metres, a front distance of 5 metres and a colour of mid grey (0.5, 0.5, 0.5). If the sphere is positioned so that its centre is 10 metres away from the viewer then what colour does the closest point on the sphere appear to the viewer?

(iii) What are the texture coordinates at the top right hand corner of a texture image with a width of 128 pixels and a height of 64 pixels. Explain your answer.

(iv) How do you identify the front face of a polygon? Give an example of front-facing and back-facing polygons. Why does Java 3D differentiate between front-facing and back-facing polygons?

(v) What is the difference between direct volume rendering and indirect volume rendering? Give examples of each approach to support your answer.

(vi) If an ambient light with a cyan colour (0.18, 0.80, 0.87) illuminates an object with a purple ambient colour (0.65, 0.23, 0.66) what colour will the object appear to be in the rendered scene?

(vii) What is the difference between the **`SCREEN_DOOR`** and **`BLENDED`** transparency modes defined by the **`TransparencyAttributes`** appearance component?

[14 marks]

(b) A line is defined by its two endpoints. Using the equation below rotate the 2D line connecting (-3, -7) to (2, 2) by 30 degrees about the origin. Note that a positive angle represents anticlockwise rotation about the origin.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

[5 marks]

Question 1 continued overleaf…

(c) What are the two rules that relate to the definition of polygons in VRML? Describe in detail the operation of the VRML code listed below and provide a wire frame illustration of the expected outcome.

```
#VRML V2.0 utf8

Shape
{
    appearance Appearance
    {
        material Material
        {
            diffuseColor 0 0 1
        }
    }
    geometry IndexedFaceSet
    {
        coord Coordinate
        {
            point [ -1 -1 -1,
                     1 -1 -1,
                    -1  1 -1,
                    -1 -1  1,
                    -1  1  1,
                     1 -1  1,
                     1  1 -1,
                     1  1  1 ]
        }
        coordIndex [ 0, 3, 4, 2, -1,
                     0, 1, 5, 3, -1,
                     0, 2, 6, 1, -1,
                     7, 5, 1, 6, -1,
                     7, 6, 2, 4, -1,
                     7, 4, 3, 5, -1 ]
    }
}
```

[6 marks]

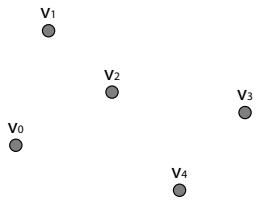
Question 2

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

- **`BranchGroup`**
- **`OrderedGroup`**
- **`TransformGroup`**
- **`ViewSpecificGroup`**

[8 marks]

(b) Use the following set of vertices to define a line strip array, triangle strip array and a triangle fan array. A single strip should be generated for each type of geometry and each of the triangles should be numbered in the case of the triangular geometries.



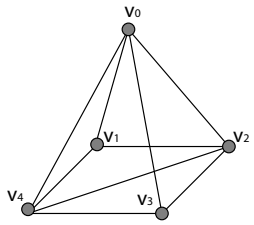
What is the main benefit of using strip geometry over regular geometry?

[4 marks]

Question 2 continued overleaf…

(c) Describe, in relation to the piece of geometry illustrated below, the difference between the following subclasses of **`GeometryArray`**:

- **`TriangleArray`**
- **`IndexedTriangleArray`**



In both cases, calculate the number of bytes required to store each version of the geometry. Given that a Java **`float`** primitive requires 4 bytes and a Java **`int`** primitive requires 4 bytes. Assume that the vertices are represented using **`Point3f`** objects.

[5 marks]

(d) Draw the **`BranchGroup`** rooted scene graph represented by the code listed below. Explain the operation of the code with reference to the scene graph 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.    Switch switchGroup = new Switch(Switch.CHILD_MASK);
 6.    BitSet mask = new BitSet(3);
 7.    mask.set(0);
 8.    mask.set(2);
 9.    switchGroup.setChildMask(mask);
10.    root.addChild(switchGroup);
11.
12.    SharedGroup sharedGroup = new SharedGroup();
13.    ColorCube cube = new ColorCube(0.2);
14.    sharedGroup.addChild(cube);
15.
16.    Transform3D t1 = new Transform3D();
17.    t1.setTranslation(new Vector3f(-0.25f, -0.25f, 0.0f));
18.    TransformGroup tg1 = new TransformGroup(t1);
19.    Link link1 = new Link();
20.    link1.setSharedGroup(sharedGroup);
21.    tg1.addChild(link1);
22.    switchGroup.addChild(tg1);
23.
24.    Transform3D t2 = new Transform3D();
25.    t2.setTranslation(new Vector3f(0.25f, -0.25f, 0.0f));
26.    TransformGroup tg2 = new TransformGroup(t2);
27.    Link link2 = new Link();
28.    link2.setSharedGroup(sharedGroup);
29.    tg2.addChild(link2);
30.    switchGroup.addChild(tg2);
31.
32.    Transform3D t3 = new Transform3D();
33.    t3.setTranslation(new Vector3f(0.0f, 0.25f, 0.0f));
34.    TransformGroup tg3 = new TransformGroup(t3);
35.    Link link3 = new Link();
36.    link3.setSharedGroup(sharedGroup);
37.    tg3.addChild(link3);
38.    switchGroup.addChild(tg3);
39.
40.    return root;
41.  }
```

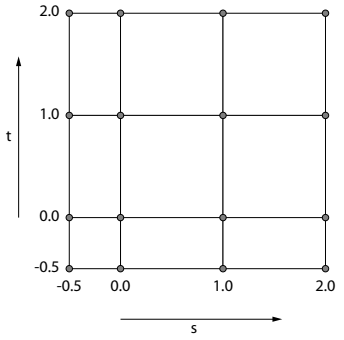
[8 marks]

Question 3

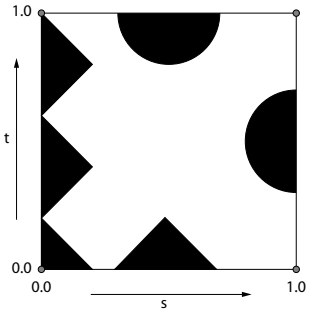
(a) Describe the approach to texture mapping implemented by Java 3D. What is the difference between the following two boundary modes defined by the **`Texture`** class?

- **`CLAMP`**
- **`WRAP`**

Illustrate the effect of texture mapping on the following piece of geometry if the horizontal boundary mode is set to **`CLAMP`** and the vertical boundary mode is set to **`WRAP`**. Use the texture image that is provided in your illustration.



3D geometry with 2D texture coordinates

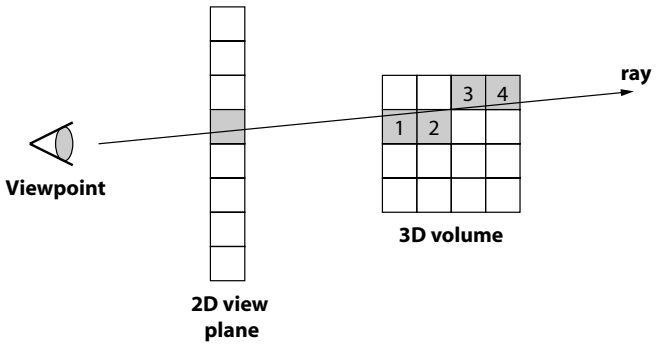


2D texture Image

Describe one method provided by Java 3D for automatically generating texture coordinates from geometry.

[9 marks]

(b) Describe the operation of the ray casting approach to volume rendering. The image below is a cross-sectional representation of the ray casting process.



Calculate the value of the shaded pixel in the 2-D view plane using this equation:

$$c_{out} = c_{in}(1-\alpha) + C\alpha$$

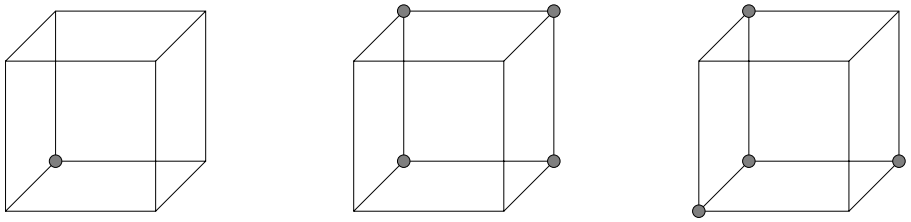
The colour and opacity values for the voxels numbered one to four are:

Voxel	Colour (c)	Opacity (α)
1	18	0.06
2	15	0.03
3	228	0.92
4	42	0.13

[8 marks]

Question 3 continued overleaf…

(c) 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 4

(a) Describe the Binary Space Partitioning (BSP) approach and the use of BSP Trees as space subdivision management structures.

[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. (Note: you can trace this figure through to your answer book page)

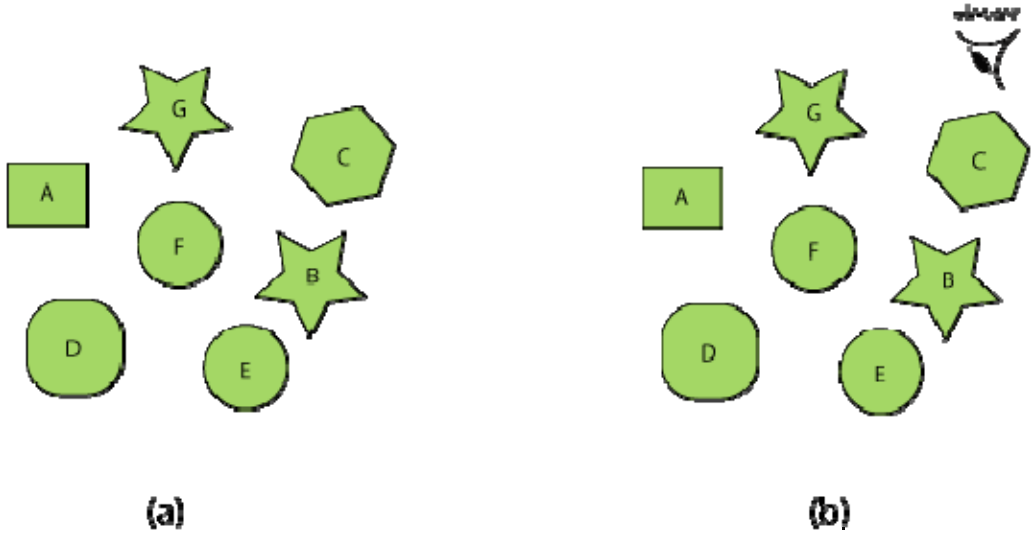


Figure 4.1 (a) Example scene with objects A to G (b) illustration of a viewer in the top right-hand corner, facing towards the centre of the object cluster (object F).

[6 marks]

(d) 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 4 continued overleaf…

(e) Compare and contrast the BSP approach to the Bounding Volume Hierarchy (BVH) approach. How does the use of these approaches differ?

[4 marks]

Question 5

(a) OpenGL has two matrix modes GL_MODELVIEW and GL_PROJECTION. Describe the use of these two matrix modes.

[4 marks]

(b) 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.

[8 marks]

(c) Draw and describe the output of this segment of C++ OpenGL code:

```
void drawSomething(GLfloat dia)
{
    GLfloat colours[8][3] = {
        {1.0, 0.0, 1.0}, {1.0, 0.0, 0.0},  //magenta ,red
        {1.0, 1.0, 1.0}, {1.0, 1.0, 0.0},  //white, yellow
        {0.0, 1.0, 1.0}, {0.0, 1.0, 0.0},  //cyan ,green
        {0.0, 0.0, 1.0}, {0.0, 0.0, 0.0}   //blue, black
    };
    GLfloat vertices[8][3] = {
        {-dia, dia,  dia}, {-dia, -dia,  dia},
        { dia, dia,  dia}, { dia, -dia,  dia},
        { dia, dia, -dia}, { dia, -dia, -dia},
        {-dia, dia, -dia}, {-dia, -dia, -dia}
    };
    int strip[6] = {2, 0, 1, 7, 6, 0};
    int bottom[8] = {0, 1, 7, 6 };

    glBegin(GL_TRIANGLE_FAN);
      for(int i=0; i<6; i++)
      {
        glColor3fv(colours[strip[i]]);
        glVertex3fv(vertices[strip[i]]);
      }
    glEnd();

    glBegin(GL_QUADS);
      for(int i=0; i<4; i++)
          {
              glColor3fv(colours[bottom[i]]);
              glVertex3fv(vertices[bottom[i]]);
      }
    glEnd();
}
```

[8 marks]

(d) What functions does the OpenGL stack perform? Use a short segment of pseudo-code to describe how you would use this stack to place objects in a scene.

[5 marks]

Question 6

(a) Explain the following terms, which are used to describe shaded surfaces: *Specular*, *Translucent*, and *Diffuse*.

[3 marks]

(b) Describe the Phong Shading Model. What is the Lambertian Surface Model and how does the intensity of reflected light vary as the angle to the light and viewer changes with respect to the surface normal?

[7 marks]

(c) 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.

[10 marks]

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

[5 marks]