THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO3355 ZB**

BSc Examination

COMPUTING AND INFORMATION SYSTEMS AND CREATIVE COMPUTING

Advanced Graphics and Animation

Date and Time: Thursday 19 May 2016, 10.00 – 12.15

Duration: 2 hours 15 minutes

There are FIVE questions on this paper. Candidates should answer **THREE** questions. All questions carry equal marks and full marks can be obtained for complete answers to **THREE** questions. The marks for each part of a question are indicated at the end of the part in [.] brackets.

Only your first **THREE** answers, in the order that they appear in your answer book, will be marked.

There are 75 marks available on this paper.

A hand held calculator may be used when answering questions on this paper but it must not be pre-programmed or able to display graphics, text or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

1.

(a) Name and briefly describe the five stages involved in a typical graphics pipeline, going from geometric models to images.

[7]

(b) Based on matrix operations,

(i) provide a mathematical expression that rotates a vector $[x, y]^T$ in the 2D plane by θ degrees.

[1]

(ii) provide expressions to rotate a 3D vector $[x, y, z]^T$ by θ degrees about each of the three axes.

[3]

(c)

(i) Construct the matrices and write the expression that translates a vector $[x, y, z]^T$ by [5, 2, 1] and then scales the result by a factor of 0.3.

[4]

(ii) Compute the coordinates of $[1, 4, 2]^T$ after translation by [5, 2, 1] and scaling by 0.3.

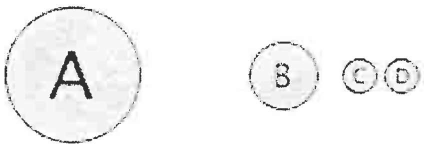
[2]

(d)

(i) Briefly describe what is meant by the term *transformation stack* and how it can assist in constructing hierarchical models of objects.

[3]

(ii) Consider the example of a simple two-dimensional solar system, consisting of four planets, named A, B, C and D respectively, as depicted below:



Suppose that you want to create an animation where A remains static at the center of the frame, B rotates around A, while both C and D rotate around B, but in opposite directions. Provide pseudocode for an iterative function –such as Processing's `draw()` – that draws the planets and animates them. Briefly explain how you make use of the transformation stack.

[5]

2.

(a) Explain what is meant by
   (i) *Object Coordinate System*
   (ii) *Local Coordinate System.*

[4]

(b) Explain the motivation and the basic idea behind the z-buffer algorithm. Describe its steps, providing pseudocode. Can you identify a drawback of its use?

[8]

(c) Name three methods for filling a closed polygonal region with a single colour.

[3]

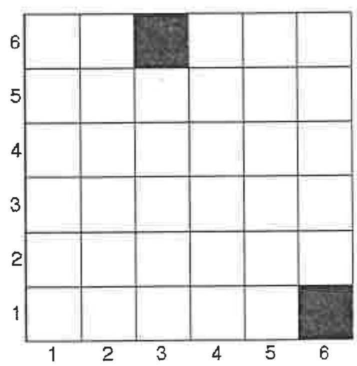
(d)
   (i) Describe the steps of the Bresenham line drawing algorithm for the case of a line with a positive slope of less than 45 degrees.

[4]

   (ii) How would the former be modified in order to treat a line with a slope between 90 and 135 degrees?

[2]

(e) Using the Bresenham line drawing algorithm, calculate the pixels to draw a line between points (6,1) and (3,6) as per the illustration below. Show your working.



[4]

3.

(a) Explain what is meant by *triangle strips* and *triangle fans*. How do they differ from each other and what is their advantage over individual triangles in Processing?

[6]

(b) Explain what is meant by 24-bit RGB colourspace. How many colours can it represent? How is 32-bit RGBA different?

[4]

(c) What is image-based lighting and how does it compare to environment mapping? Explain how the process is related to global illumination and high dynamic range imaging and describe the steps involved.

[8]

(d) The following listing is the source code of a GLSL shader. Identify the type of shader and explain what it does and how it does it, line by line.

```
varying vec4 vertColor;
varying vec3 transformedNormal;
varying vec3 lightDir;
void main() {
       vec3 direction = normalize(lightDir);
       vec3 normal = normalize(transformedNormal);
       float light = max(0.0, dot(direction, normal));
       gl_FragColor = vec4(light, light, light, 1) * vertColor;
}
```

[7]

UL16/0096

4.

(a) How does texture mapping differ from procedural texturing?

[4]

(b) Name three types of map shapes and the coordinate system used to implement each one of them. Describe how mapping takes place between the coordinates of the texture map and the object in each case.

[6]

(c)

(i) Describe the technique of *displacement mapping*, explaining how it affects the object geometry and the number of polygons.

[4]

(ii) How is displacement mapping different from *bump mapping*? Taking into account the complexity of each technique and the quality of representation it achieves, state which of the two you would choose in order to represent a field of grass as seen from a satellite in the sky. Would your choice change should the grass be seen in a close-up shot from the ground? Justify your answer.

[6]

(d) What is described with the term *environment mapping*? Explain the steps of the process.

[5]

5.

(a) In the scope of physics modelling, explain the following terms:

(i) *Particles*

(ii) *Rigid bodies*

(iii) *Compound bodies*

(iv) *Soft bodies*

[8]

(b) BRigid provides classes to construct simple (primitive) and compound collision shapes. Name an appropriate class for each.

[2]

(c) Which types of friction exist in physics? How are they treated by physics engines? Which command is used to control their coefficients in BRigid?

[6]

(d) What does the following code do? Explain how it does this line by line.

[9]

```
public void setup() {
  size(1280, 720, P3D);
  physics = new BPhysics();
  physics.world.setGravity(new Vector3f(0, 500, 0));
  BSphere s = new BSphere(this, 2, 0, 0, 0, 20);
  s.rigidBody.setDamping(0, 0);
  Vector3f pos = new Vector3f(640, 0, 0);
  BObject r = new BObject(this, 100, s, pos, true);
  physics.addBody(r);
}

public void draw() {
  lights();
  physics.update();
  physics.display();
}
```

END OF PAPER