Examiners' commentaries 2016–17

CO3355 Advanced graphics and animation – Zone B

General remarks

The general standard of performance on this examination paper rose compared to the previous year. There were a significant number of strong papers, with several candidates achieving marks of over 90 per cent.

While most candidates could demonstrate appreciation of the relevant issues in the questions, some struggled to provide clear and concise explanations and as a result lost marks. Providing succinct and accurate explanations is worth focusing upon in revision and examination practice, and should build upon skills developed in earlier assignments and examinations.

Question 3 was most popular with 94 per cent of candidates attempting it, closely followed by Question 1 with 85 per cent. 56 per cent of the candidates answered Question 4, while Questions 5 and 2 were less popular with 38 per cent and 26 per cent respectively.

Question 5 attracted the best answers, followed by Questions 2, 1 and 3 (in that order). Question 4 attracted the lowest marks.

Comments on specific questions

Question 1

a. This question required a very basic knowledge of related mathematics and the majority of candidates answered it well. However, some incorrectly considered the result to be a vector and not a scalar.

b. When two vectors are orthogonal their dot product will equal to zero, which in this case yields $a_1b_1+a_3b_3 = 0$. Unfortunately, most answers here were either vague or not relevant.

c. This matrix represented a scaling by [3, 1, 1] followed by a translation by [2, 2, 2]. The result of the calculation when applied to the given vector was: $[5, 4, 2.5, 1]^T$.

   The majority of candidates managed to calculate the result correctly. However, many had issues with explaining the type of transformations performed. This is dependent on a very basic understanding of the transformation matrix elements that indicate S, R and T respectively. Knowledge of the transformation matrix structure leads easily to the correct interpretation.

d. This part of the question examined the effect of order when performing geometric transformations. Candidates were required to assess whether certain propositions were true or false and to justify their answers. Regarding the specific propositions:

   i. was false, as in both cases the object moves.

   ii. was true, as the scaling matrix was the same.

   iii. was false, as scaling when performed after translation will modify the latter and translate the object more.

   iv. depended on the scaling factor. If greater than 1 (the object getting larger) then it would be translated more in case b and the proposition would be false. Otherwise the proposition would be true.

Marking was fairly flexible, accommodating some answers that made other assumptions, as long as they were reasonable and well-explained. However, some candidates had problems providing satisfactory explanations, even where they identified the validity of each proposition correctly.

e. This part of the question asked candidates to compare perspective and parallel projections by identifying an advantage of each and stating how a set of parallel lines would appear in each case. The vast majority of candidates answered this correctly, though a few confused the two transformations.

Question 2

a. This part of the question examined *keyframing* in traditional animation, asking candidates to identify the problem it solves and the way it does it. This was essentially bookwork, covered in Chapter 8 of the subject guide, and was mostly well answered.

b. This section required candidates to identify that interpolation helps to automatically calculate in between frames and achieve a smooth motion. Candidates answered this question well.

c. This part of the question tested candidates' understanding of linear interpolation. A key frame was used for every 10 frames, using a formula such as P(t) = (1/10)* (t*P(tA) + (10-t)*P(tB)), with t=9 yields the result: [8.7, 4.9, 2.8]. Some candidates were confused and did not use linear interpolation or only calculated the middle frame between the two keyframes. Nevertheless, this was answered well by most candidates.

d. This part of the question was about skeletal animation, asking candidates to describe how bone position is calculated using forward kinematics and to derive the corresponding formula for the given figure. The former was essentially bookwork, while for the latter a formula of the form $R_1 (R_2(L_2)+L_1)$ would suffice.

   Many candidates solved the problem analytically, i.e. first calculated the formula for the first joint: (x1,y1)= (sinR1 * L1, cosR1 * L1) and then for the second: x2 = (sin(R1+R2-180)*L2) + x1, with y2 = (cos(R1+R2-180)*L2) + y1. This answer is also correct and attracted full marks. Generally, most candidates provided satisfactory answers here.

e. This question required the application of the law of cosines to solve an instance of the inverse kinematics problem presented in (c). Good answers included verification that the desired position was within reach and then solved the expression to calculate the cosine of each of the three angles, and then the angles themselves. Most candidates answered this question well.

Question 3

a. This part of the question required candidates to describe the purpose of a *local illumination* model. Most candidates provided good answers, though others were not very clear, failing to identify the distinction between direct and indirect light.

b. Good answers to this part of the question explained that flat shading computes lighting values for one vertex and uses the resulting colour for the entire polygon, which then results in a single flat colour. Interpolated methods use interpolation, constructing new data points within the range of a discrete set of known data points (for colour values or surface normals) to provide a much smoother result. Gouraud and Phong are methods of interpolated shading. The vast majority of candidates identified methods well but some had problems describing flat shading.

c. This part of the question required explanation of the concept of *radiosity*. This was essentially bookwork, covered in section 7.8 of the subject guide. Most candidates demonstrated a good understanding of the concept.

d. This was a simple exercise to test the candidates' understanding of the Phong lighting model. It provided a diagram consisting of a light source, and three cameras pointing on a surface, asking which one sees the brightest (i) diffuse and (ii) specular illumination. As diffuse reflection does not depend on the angle of view, it does not change among the cameras. On the other hand, Camera A would see the brightest specular reflection, as the angle from the point to the camera was closest to the angle from the light to the point.

   About 50 per cent of candidates were confused by (i), while almost all provided the correct answer in (ii). However, many found it difficult to justify their choice.

e. This part of the question asked for the functionality of the provided shader code together with a description of its main steps. The code is a fragment shader that showed a fragment colour varying with time. It used a 'time' variable and a sine function to choose the R colour component and then exchanged the G and B components of the vertex colour. The result was the composite fragment colour. The majority of candidates did not interpret the shader well. Some candidates failed to acknowledge that y and z are the G and B channels respectively. Instead they thought they were coordinates.

Question 4

a. This part asked candidates about the concept of *aliasing* and how it is related to texture mapping. Although most candidates identified that procedural texturing is not affected, most failed to provide a clear explanation of the concept.

b. This part of the question was about *Perlin noise*, a technique used to generate noise that appears more natural. For this, the noise ( ) method in Processing is used, rather than the more familiar random ( ) method. Though most candidates identified the notion, many did not mention the Processing function but elaborated how it can be used in general (such as for implementing bump mapping, for example). Such answers also attracted some marks.

c. This part of the question compared the techniques of texture mapping and bump mapping in terms of achieved realism in three different situations when lighting is modelled in the scene. In all three conditions, bump mapping helps produce more realistic results. Very few candidates provided clear answers to this one.

d. i. In this part, the code of a vertex shader that performs simple texture mapping was given, and candidates were asked to explain what it does. The code transforms the texture coordinates of each vertex, first rescaling them and taking into account texture inversion along the Y-axis and the non-power-of-two textures, passing the transformed coordinates to the fragment shader. This was mostly well-answered.

   ii. This part asked candidates to consider the corresponding fragment shader, and fill in the missing commands. The answer could be given in one command such as:

   ```
   gl_FragColor = texture2D(texture, vertTexCoord.
   st) * vertColor;
   ```

   Only a few candidates provided this correctly.

Question 5

a. This part of the question asked candidates about the notion of *uniform* variables in GPU programs, requiring three examples of their use. Most candidates provided good answers, accompanied with appropriate examples.

b. This part required a drawing to illustrate a situation where Painter's algorithm would fail to represent correctly. This was mostly well answered.

c. This part of the question examined the use of *BSP trees* in the context of the *Z-buffer*. This was mostly well-answered. Some candidates described the functionality of the Z-buffer algorithm in detail; this was not necessary to attract full marks.

d. This part of the question asked candidates to state whether the use of a fragment or a vertex shader would be more appropriate for four different tasks, justifying their answers. A fragment shader is appropriate for image histogram equalisation as this requires pixel manipulation. A vertex shader is suitable for skinning, Gouraud shading and displacement mapping that involve transformation of vertices. Only a few candidates managed to provide correct answers with clear justification.