Examiners' commentaries 2016–17

CO1112 Creative computing I: image, sound and motion – Zone A

General remarks

Overall performance on this paper was weak compared to previous years, with the average mark being a high third class. A larger proportion of candidates than usual failed, and there were fewer firsts (around 10 per cent), though the odd candidate did extremely well, obtaining almost full marks. This year again, all candidates answered exactly four questions, which meant they did not lose marks unnecessarily for leaving out questions or for answering too many.

What follows is a brief discussion of the individual questions on this paper, with hints and explanations of the answers expected by the examiners.

Comments on specific questions

Question 1: General

This year, Question 1 was made up of several short questions covering general topics across the course. This was one of the two most popular questions, answered by more than two thirds of candidates. Performance on this question was among the highest overall, with the average mark being in the lower second range. Very few candidates obtained more than 20 marks out of a possible 25, which was disappointing, but in line with the overall performance on this paper.

Most candidates were able to answer part (a) well, explaining that a digital image is constructed of numerous pixels, depending on resolution. Other relevant information would include that each pixel is a single datapoint containing information about the colour/transparency at each point in the image. The RGB and Alpha values are stored.

For part (b), almost all candidates understood that JPG and PNG are stored in basically the same way, and the different formats are about the kind of compression, and fidelity to the original image.

For part (c), explanation of what pixellation is was sometimes weak, or even absent. It is important to answer all parts of a question. Candidates lost marks for not explaining how to mitigate against it occurring (for example, using a higher resolution for digitising and storing the image), and discussion of whether it is desirable (which it certainly can be) was often omitted.

Part (d) accepted all reasonable answers that indicated understanding; digital audio is the data that we use to represent sound on a computer.

Part (e) was often not answered particularly well, showing confusion between digitisation of sound, and compression of sound. Many candidates included excessive information on the Nyquist frequency, which is not centrally relevant to the subject of storing sound, thereby showing limited understanding of the concepts. Additionally, information about encoding itself was often missed. To store sound digitally, there are two essential aspects that affect the fidelity of the representation: sample rate (when the analog signal is measured at a variety of regular points and converted to a digital representation); and bit depth (which refers to the degree of accuracy with which each reading is represented). With regard to storing sound, files can be compressed to save space; this is a different issue to representation, and not all candidates understood this.

For the final part, a simple response that a Boolean data type is one that can take on the values true or false (and only true and false), was expected. Other answers that showed understanding were accepted. Note that the values 0 and 1 are not Boolean values, even though the Boolean values false and true may be represented digitally using only zeros, or only ones.

Question 2: History and Creative Thinking

This was one of the least popular questions, answered by less than half the candidates, but with reasonable marks being achieved. Only a few candidates who chose this question failed to achieve a pass mark for it.

For part (a), a simple response was expected: Pythagoras' Theorem states that the sum of the squares of the two sides of the right angles of a triangle is the same as the square of the hypotenuse. Some candidates failed to mention that the triangle must be a right-angled one, but the question was generally well answered.

The calculations for the sketch for part (b) are in the subject guide, and creating the sketch was straightforward.

Part (c) showed the most variety; examiners were expecting a description of some of the underlying Bauhaus philosophy, and then discussion of which aspects are amenable to computation. Some candidates mentioned design, architecture, Itten's three basic forms, for which *Processing* can be seen as a workbench. There were other well-constructed responses. In general, marks were awarded where the description of the aspects of the philosophy was clear, and the connection with computational technology was explicitly made.

There were many strong responses for part (d). Answers that scored higher marks linked the cultural context to how the image can be understood, as well as commenting on and discussing the image.

Question 3: Colour

This was a very popular question, answered by almost all candidates, but with a very weak overall performance. The average score on this question was barely a pass, with the highest mark 18 out of a possible 25. Candidates are reminded to be self-reflective about what they do or do not know, and to choose questions that highlight their strengths. It is essential to read through the whole question before making this decision.

A good answer for part (a) would include descriptions of both what RGB is and what HSB is, and their differences. Noting that colour changes in HSB are related to hue changes, which are what we perceive, is important in explaining why HSB is seen as being closer to how humans perceive colour. Other appropriate comments which attracted marks include that saturation is about intensity (for example, the kinds of colours children traditionally like), as well as information about the differences in use; for example, RGB and use in computer monitors, and HSB for palette selection.

The first question in part (b) is relatively staightforward, requiring a description of the sketch behaviour: six rectangles with ellipses inside them are drawn, each containing a pair of ‘opposite’ colours. The ‘opposite’ is obtained through the bitwise not.

The second part is more complex, because the concept of ’inverse’ colours in RGB is not as straightforward as it is in the HSB model. Not many candidates explicitly commented on this. Any approaches to somehow dividing the space into threes that is reasonable was accepted, though most candidates struggled with this.

For part (c), answers were also generally weaker. A monochrome palette is one that uses a limited or restricted part of the colour wheel. Greyscale is an extreme version of this that uses only black and white; however some candidates confused greyscale and B&W palettes. Sepia is another, but any bands of a colour model can be used. A monochrome can be very effective and is also cheaper to print; some good examples of effectiveness were given.

Question 4: Generative systems

A popular question, answered by many candidates, and with average results.

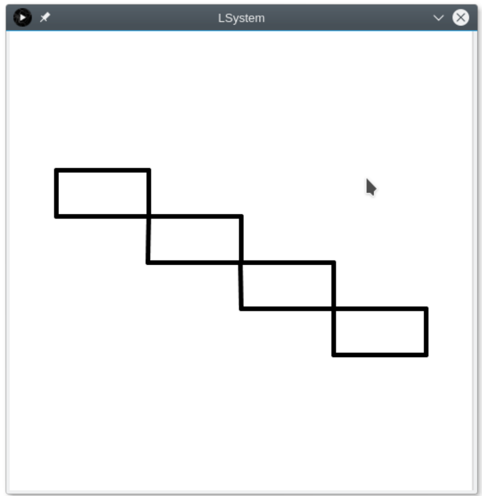
Almost all candidates correctly answered part (a): the initiator is “HH” and there are two generators, “F→H+F+” and “H→F-H-”.

Part (b) was also answered reasonably by most candidates, saying that the three calls result in the following:

```
1: F-H-F-H-
2: H+F+-F-H--H+F+-F-H--
3: F-H-+H+F++-H+F+-F-H---F-H-+H+F++-H+F+-F-H---
```

In contrast, part (c) was very poorly done, with only the odd candidate providing the correct sketch, along the lines below. Because of this, few candidates correctly identified that the coordinate of the top-left point was (50,150) and the bottom-right point was

(450,350).



Part (d), generally reasonably done, required the use of the `default` keyword and correct code to do the drawing and translation.

For part (e), correct use of `pushMatrix()` and `popMatrix()` was required, and again, this was done generally well.

Part (f) saw some reasonable answers. The descriptions of mutation were adequate, and good answers included the fact they are one of the key mechanisms by which variety is introduced into the population. For the second sub-part, examiners expected the answer to deal with substituting a single character in the string for another one drawn from the legal set of characters, inserting a new character at a random point in the string, and deleting an existing character at random. A complete answer would include a discussion of checking that the delete operator does not result in an empty string, and also the possibility of adding a limit on how long the string might be allowed to grow.

Question 5: Motion and interaction

This question was the most popular of all, answered by almost all candidates, and with the best average performance.

Part (a) was not answered well by many candidates, who instead of describing the overall behaviour of the sketch, instead just gave a blow-by-blow decoding of what was being done. What was expected was an account of a circle of diameter (not radius) 30 pixels, drawn with a black circumference of width 3 pixels, and filled with red on a white background. It moves with Perlin noise around the screen, leaving a trace of its previous path, which gradually fades away. Stating what `translate(), fill()` and `ellipse()` do, for example, is not describing the behaviour of the sketch.

For part (b), almost all candidates were able to explain correctly that in line 14, 50 specifies the alpha value, while in line 21, 0 specifies the green channel value.

Part (c) comprised three parts, all of which were reasonably answered. Most candidates knew that `random(n)` returns uniform-random floats between 0.0 and n, while in contrast, `noise(x)` produces more 'natural' variation, and x defines a position on a predefined noise function. They were also able to explain that the effect of the changed code in the second sub-part would be to restrict the circle to moving on a diagonal line from top-left to bottom-right of the screen. Finally, most candidates were able to say that the change in the third sub-part would make the movement of the circle more random and jumpy; some clarified further that this is because calling noise with bigger differences between each call results in larger differences in return value.

The simplest answer for part (d) is to replace `fill(255, 50)` in line 14 with:

```
fill(mouseY * 255 / height, 50);
```

All appropriate responses were awarded marks.

For part (e), candidates needed to add a global Boolean variable to track current state, a `mouseClicked()` function to toggle shape, and to make the drawing of the shape on line 22 conditional on the state of the variable. Some candidates did not centre the square properly on `x,y`, which could be done using appropriate arguments to `rect()`, or by setting `rectMode()`).

Question 6: 3D graphics and effects

This was the least popular question, and many candidates gave weak answers. The average mark for this question was 10 out of a possible 25 marks, though the odd candidate did obtain close to full marks.

Part (a) was generally well-answered; the first three parameters are the x, y, z position of camera; the second three are the x, y, z of the centre of screen (focus point); and the final three specify which axis is facing up.

Few candidates produced a good output sketch in response to part (b). Examiners were looking for the correct perspective with respect to the camera position, correct dimensions and positions of boxes, and colours. The camera is level with the centre of the middle box, looking at it rotated 45 degrees (which appears visually as on to one of the vertical edges).

For part (c), the simplest solution would be the following:

```
rotateY(TWO_PI * (frameCount/15.0));
```

Some incorrect responses did not take the `frameRate` of 15 into account to get the correct angular velocity, though most candidates were able to obtain the other aspects correctly.

Part (d) required candidates to know that it is possible to move the camera to achieve the same visual effect as rotating the boxes. This could be achieved with calls to `camera()` in the `draw()` function, while updating the camera position (the first three parameters) at each call to describe a circular path around the boxes. Alternatively, a more sophisticated approach would be to use camera transformations (as described on p.72, subject guide, Vol. 2), using `beginCamera(); rotateX(a); endCamera();`. Either approach was awarded full marks if correctly described, though few candidates chose the second approach.

Many candidates did not answer part (e), which is straightforward. Translate can take three parameters for each axis, because its operation is independent for each axis. However, application of rotation about different axes is not independent; the end result depends on the order in which the rotations are applied. Hence, the rotation functions are separated into their separate axes.

The first sub-part of (f) was answered well by most candidates; the transformation represents a translation of (2, 1, 3) along the x, y and z axes respectively.

The second sub-part proved more challenging, and descriptions were weak. While having an extra row in the transformation matrix makes no difference to the calculations for x', y' and z', it makes the transformation matrix a square matrix with dimension equal to the size of the column vector representing the original point. This allows us to combine a sequence of matrix multiplications, representing a sequence of transformations, into a single expression.

Responses to part (g) were limited, where candidates expressed the transformation by decoding the mathematical operations, rather than using words. The transformation represents a scaling of 2 along x and y axes (with no scaling on the z axis), followed by a translation of (2, 1, 3) along the x, y and z axes.