Examiners' commentaries 2016–17

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

General remarks

Overall performance on this paper was reasonable, with the average mark being a medium second class. The odd candidate did extremely well, obtaining almost full marks, and around a quarter of candidates obtained a first-class mark for the examination. 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 least popular questions, answered by less than half of candidates. Performance on this question was average, though the odd candidate obtained an impressive almost full marks out of 25.

Most candidates were able to answer part (a) well, explaining that each pixel is a single datapoint containing information about the colour/transparency at each point in the image. It was important to mention explicitly that the RGB and Alpha values are stored.

For part (b), almost all candidates understood that the value can be obtained from either using PImage.width and PImage.height, or using the length of the Pimage.pixels[ ] array. Not all candidates included a code fragment as required, resulting in a loss of marks.

For part (c), most candidates were able to explain that texture mapping is the process of laying a texture onto a 3-D drawing or image.

Part (d), following on from part (c) required at least the mention of the *Processing* method `texture()`, as well as `beginShape()` and `endShape()`.

Part (e) was a straightforward question, from material in the subject guide, and usually candidates responded well, saying that digital audio is the data we use to represent sound on a computer.

Part (f) was often not particularly well answered, with many candidates only describing sampling, and not discussing encoding at all. 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. Finally, examples were not always given.

For part (g), 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.

Most candidates were able to give appropriate examples for the final part; however, it was important that the examples were functions, rather than simply methods.

Question 2: Colour

This was a very popular question, answered by many candidates, and with a good overall performance. The average score on this question was good, with a handful of candidates obtaining more than 20 out of the possible 25 marks.

A good answer for part (a) would include descriptions of both what RGB is and what HSB is, and their differences. Examiners then expected an explanation of what a colour scheme is. It is generally important to mention the concept of the colour wheel; an example of a colour scheme could be, for example, the use of colours opposite in the wheel. In terms of contrast between RGB and HSB, a colour scheme is easier with HSB because it is more similar to the way human beings perceive colour (and a colour scheme is really a cognitive and perception thing), rather than a composite of red, blue and green. Other relevant points were awarded marks.

Part (b) was straightforward, asking about simple *Processing* aspects. `stroke()` sets the colour of the line, while `fill()` sets the colour of any filling that is then done. For the second sub-part, examples as well as descriptions were expected.

The first sub-question in part (c) is relatively straightforward, requiring a description of the sketch behaviour: the colours cycle through the colour wheel, and at each stage there are three colours, taken from equally spaced places on the wheel. Different shapes are drawn, using these colours. Some candidates included a pleasing level of detail in their descriptions, and most understood that examiners were looking for description of what happens, rather than a decoding approach that just described the effect of each line of code.

The second sub-part was not always answered well. Many candidates realised that the best approach was to pick the colours from equally spaced places on the colour wheel, and add a fourth colour variable, `c4`. However, not everyone understood that the shape drawing should be parameterised; and many also did not make use of the new variable they had added, in the code that followed. For the third sub-part, examiners expected candidates to understand that while changing to an n-colour scheme is conceptually possible, as n gets large, it becomes visually or cognitively meaningless. If you were to do it, you might make the appropriate variables (`c1`, `c2`, *etc.*), an array. It would also be useful to rewrite the shape drawing part to be more algorithmic, rather than hardcoded as it is.

Question 3: Creative thinking and computation

This was the least popular question, chosen by less than a third of candidates, with the weakest overall performance. No candidates obtained more than 15 out of the available 25 marks.

Part (a) is straightforward, and is covered in the subject guide. Students on Itten's Basic Course were taught theories of form with emphasis on the simple basic forms of circle, triangle and square. Compositions were made employing the three shapes. These shapes were derived from Cubism and were seen as historically primary in art. Additionally, they are independent of nature and easily produced, appearing in Itten's Wood and Metal Workshops.

The second part was reasonably answered; most candidates used `rect()` or `quad()` for the square, `ellipse()` for the circle, and `triangle()` for the triangle, and indicated the appropriate parameter restrictions for the square and circle.

Answers to part (c) were often very weak. Some candidates believed that Dürer's 'perspective machine', developed in the 1500s, used the concept of Artificial Intelligence — a point which could be stretched, but they did not seem to know of its true background; others said that the main concept was the vanishing point, but did not describe how it worked. The machine was made of two transparent grids — one that covered the scene to be drawn, and one over the canvas. The artist copied block by block what they saw in the grid into the same square on the canvas' grid, in a fairly mechanistic manner. It was expected that the images would be stilted, but this is not the case.

Following on, part (d) required an understanding that the machine relies on seeing and copying in a grid approach, and hence some aspect of this would need to be incorporated. Once this is done, the rest is algorithmic and mostly straightforward.

Finally, for part (e), the cultural context of the Banksy artwork was originally the Calais jungle, and refugees trying to get to the UK. It may be relevant to the current context, with politicians in the USA wanting to refuse entry to refugees, for example, but any other examples were acceptable. One aspect of cultural comment is that if this had happened in Jobs' case, there would be no Apple Computers.

Question 4: Generative systems

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

Almost all candidates correctly answered part (a): the initiator is "FF" and there are two generators, "F→H-F-" and "H→F+H+". Similarly, part (b) was almost always correctly answered, giving the contents as:

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

for the three calls.

Many candidates were able to draw the output correctly after three iterations, as required for part (c). There are four vertically oriented rectangles, placed corner-to-corner. It was important to note that the co-ordinate of top-left point is (150,50) and the bottom-right point is (350,450).

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, though some showed misunderstandings. 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. Though there were some good answers here, weaker attempts described visualisation aspects rather than functional ones.

Question 5: Motion and interaction

This question was the most popular of all, answered by almost all candidates, and with a good average performance, including the odd candidate obtaining full marks for their responses.

Part (a) was not answered well by some candidates, who instead of describing the overall behaviour of the sketch, instead just gave a blow-by-blow decoding of what was being done. Stating what `translate(), fill()` and `ellipse()` do, for example, is not describing the behaviour of the sketch. Good answers explained that there is a square of side length 30 pixels, drawn with a white outline of width 3 pixels and filled with green on a black background. It moves with Perlin noise around the screen. As it moves, it leaves a trace of its previous path, which gradually fades away.

For part (b), almost all candidates were able to correctly explain that in line 14, 50 specifies the alpha value, while in line 21, 200 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. Some candidates said that `random()` does more mechanical calculations, which is not as insightful. Most candidates were able to explain that the effect of the changed code in the second sub-part would be to restrict the square 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 square 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(0, 50)` in line 14 with:

```
fill(mouseX * 255 / width, 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.

Question 6: 3D graphics and effects

Candidates performed best on this question, with an average mark in the upper second; it was a popular question, chosen by more than two thirds of candidates.

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.

For 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/60.0));
```

Some incorrect responses did not take the `frameRate` of 60 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 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.

Part (e) is a straightforward question, and was answered correctly by most candidates. 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 (3, 1, 2) along the x, y and z axes respectively.

The second sub-part was also answered well in general: 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.

Part (g) was generally answered correctly. The transformation represents a scaling of 2 along the y and z axes (with no scaling on the x axis), followed by a translation of (3, 1, 2) along the x, y and z axes.