Examiner's report 2016-2017
CO2227 Interactive multimedia
Coursework

This year's courseworks involved practical implementation of algorithms that form part of the subject material for this course. For both courseworks, in-depth investigation of a particular algorithm was required, and followed by the development of an artefact based on the investigation. In the case of the first coursework, the area of consideration was distance metrics; students were required to discuss the Levenshtein algorithm, develop an implementation of it for a particular use instance, and then develop their own artefact which was inspired by what they had found in their investigation. This was then to be rounded off by an academic exercise to apply the distance metric work to the context of recommender systems. For the second coursework, the subject was inbetweening in the context of animation. Again, a discussion of the area, followed by the implementation of an algorithm and then artefact inspired by their investigation, was required.

Students were also asked to develop their ability to critique work — their own and others' — by providing self-reflection for each of their own artefacts, as well as critically examining the coursework 1 submission of another student.

At level 2, examiners expect students to be able to present coherent academic writing, with proper referencing and citation. In addition, insightful analysis and discussion of creative artefacts is expected, and students should be demonstrating that they are developing these abilities.

It is important to note that many students lost marks needlessly, in both submissions, for not following the instructions regarding the format for submission. These are not given as hoops for you to jump through, but they make the task of marking more straightforward for the examiners. If examiners are spending extra time looking for files that have not been correctly named, or have been placed in folders that have not been specified, this detracts from the overall work. Marks are deducted in these cases.

CW1: Distance metrics

There were five parts to this coursework, and many students made a reasonable attempt at all five parts. Unfortunately about a quarter of students lost marks through incorrect submission format; and around a fifth of students did not submit part D, thereby losing all of the marks for this section. This was a shame especially in one student who submitted excellent work for the first three parts, but then obtained zero for the fourth, bringing their mark down significantly. It is always better to submit a response for each section, even a weak one, rather than to completely omit any part.

The first part required a description and discussion of the Levenshtein algorithm, and a discussion of its potential application to music. Here, students were expected to read appropriate books and articles, as well as thoughtfully considering the musical context. Particularly good answers included clear desciptions of the algorithm itself, some extensions such as the Hamming distance and good discussion of music applications. One student discussed the use of edit distance to compare similarity of rhythm in music, as well as melodic similarities, and included appropriate citation for their work. Not all students seemed to understand that a very important aspect to the application of the algorithm to specific cases is converting the information to something encoded as a string, though the stronger submissions did this well.

The second part required the implementation of a *Processing*sketch to determine the distance between a pair of strings. A number of students developed sketches that required the input of two words, which was not actually asked for in the brief. An excellent submission noted that for country names with high similarity, such as Togo and Tonga, when given an input that is the same distance to both, such as Togna, the algorthim may choose the less intuitive option if the alphabetic order is used. The student suggested returning all options with the same distance and allowing the user to choose the preferred one.

Some students restricted their algorithms to handle an edit distance of only one; not a reasonable assumption in and most cases this choice was not justified. One student included a lot of material about the complexity calculations of recursive algorithms, which was both incorrect in itself and also not relevant to the coursework. It is essential to check the material submitted for correctness, and not to include extraneous material.

The third part of the coursework required a creative development of the earlier work, and some students showed great initiative. A particularly nice example was one that used the Levenshtein distance as the basis for a typing game, to test and improve typing speed. Another very creative sketch, also based on typing, produced different images based on the proportion of correctly spelt words entered, using colour and shape in an imaginitive way. An intriguing and impressive idea was to use names, and again produce a visual artefact that was constrained and shaped in terms of the Levenshtein distance of a name input to a list of names that had been previously stored. Grayscale and colour intensity were parameters, as well as shape and also movement.

Part four was often weakly done, with shallow investigation and showing little insight. However, these submissions were better than not including anything for this part. There were also some very insightful submissions and arguments, though a couple of these lost marks for not including any references or citation. One strong idea was to use multiple sequence alignment to compare traits between songs; another good response noted that both textual and acoustic similarities can be measured for music (textual being the metadata, for example), but felt that the acoustic comparison would be a more useful one for recommender systems. This was very well argued and justified, with the mention of, for example, Echo Nest, which uses both aspects.

For the final part, all but one student obtained the full 2% for the information about their swap partner, which is a significant improvement on previous years.

In general, the work was approached well. Most students included reasonable comments in their code, though some work was weakly coded and not commented at all.

CW2: Inbetweening

The second coursework began with a requirement that students critique another student's work, and then went on to give students an opportunity to examine another algorithm in depth Fewer students omitted sections than in the first coursework, and as a result, there was a much lower rate of students (less than 10%) who did not obtain a pass mark. There were also some excellent submissions, with around a third of students obtaining a first.

The first part, the critique, was done very well by some students, with one student obtaining full marks and another obtaining 19 out of a possible 20. Examples of this work is made available in the VLE. Those who did a good job were able to provide an insightful comment on both the technical and creative aspects of their swap partner's work. Weaker submissions suffered from some of the following: comparing the work being critiqued with the student's own submission; not providing a mark assignment as requested; giving a straightforward average of marks for each section to obtain a total (instead of a weighted average); focusing only on the essay and the technical aspects of the implementation, and not considering the creative aspect of the artefact or its impact. Some students failed to comment on the submission's own critique, even if one had been given. In contrast, good critiques commented on the quality of code as well as the quality of the discussion and self-evaluation in the submission, and made suggestions for improvements.

For Part B, students were required to present an academically appropriate description of tweening and its role in animation, in relation to algorithmic approaches to animation. Some submissions contained reasonable academic content but obtained weak marks due to lack of citation; other work was a mix of some strong demonstation of conceptual understanding but an excessively narrow focus. Very good sumissions were coherent, academically sound, and with appropriate justification of any comments made.

Part C produced some nice work, some of it more technically focused, but of a high standard. The better submissions were strong in all of the different implementations with a good analysis and discussion. Weaker discussion contained evaluations of the students' own code, rather than of the results or performance of the different algorithms. Other weaknesses included not describing criteria for comparison; or choosing inappropriate criteria, such as code features. One student used a Venn diagram for part of the comparison, which was a good idea, but then did not follow through on much detail of any evaluation.

For part D, there were also some very strong submissions; sometimes very simple ideas can be very effective. It is important to be able to produce work that has impact, as well as being technically strong, and a few students managed this to a very high degree. However, a number of submissions included only a very limited development from the ideas of part C, showing little creativity and imagination. Another weakness observed was to come up with a good idea, but to use a large proportion of code from other sources (even though acknowledged), thereby obtaining few marks for originality.