University of London International Programmes – Creative computing I : image, sound and motion Coursework assignments 2015–16

Introduction

The following coursework assignments are an opportunity for you to obtain a broader and deeper understanding of the material in this unit.

Completed coursework assignments are to be uploaded to the VLE for submission. In general:

- essays and discussions must be in the form of a PDF;
- any *Processing* sketches should be in a ready-to-run form (which may mean that you have to submit them as a zip file containing an appropriate Data folder – please take care to follow the file submission details given at the end of the coursework assignment); and
- it is essential that you comment any code that you submit.

Before you submit your files, be very sure to check that you can successfully unpack them and that they contain all of the files required; we recommend you try unpacking them in an empty directory on your computer, and try running the files in *Processing* from there to check that they work.

Please do not use any spaces or unusual characters in your filenames – stick with alphanumerics where possible, and only include underscores where explicitly directed to do so.

CO1112 Creative computing 1: Coursework assignment 1

The science and art of typography has been around for many centuries. For Coursework assignment 1, you are required to find out about two specific aspects of typography, discuss them in an analytical way, and create an artwork related to them.

Aims

The aims of Coursework assignment 1 are:

a. To develop your knowledge of typography and fonts.

b. To develop your knowledge and practical experience of programming systems that incorporate typography.

c. To give you the opportunity to combine various techniques you have learned about during the course to produce a creative artefact.

Introduction

In Coursework assignment 1, you will be examining and creating systems that work with typography. The two specific aspects of typography that you are to investigate in depth are serif and sans serif fonts; and kerning. You should read up what you can on fonts and on kerning in general, and become familiar with the terminology that is used when examining fonts. You should also find out how *Processing* implements fonts, and how to make use of the font capabilities that are provided.

There is a useful tutorial on the *Processing* website that is about typography, that you should look at; however you will need to look more widely than this too.

Using code that you obtain from elsewhere: Some of the tasks below require you to explicitly make use of code that has been written by other people, and which is provided to you as part of the coursework specification. It is a normal aspect of programming to look at and use other people's code to get inspiration on how to solve a problem, or to use portions of code to achieve part of what you are trying to do. Whenever you do this, you *must* include a comment in your code to say which part has been copied, and where it came from. This is good practice whenever writing code, to acknowledge the original author.

For the final task in this coursework assignment, you will not be explicitly developing from other people's code. It is acceptable to make use of existing code for inspiration, and again, if you do so, it is essential to comment this as such. Note, however, that you will get *much* more out of this coursework assignment if you write those parts of the code yourself (after reading the appropriate sections of the subject guide and other sources), rather than copying code from elsewhere.

Part 1) Background reading

This part has two aspects that you are required to read about and discuss. It is up to you to decide how to structure your answer, but bear in mind that simply summarising what you find about a particular topic, in a serial way, is not always the best approach to take. Pay attention to proper approaches to academic writing in your submission.

Find out what kerning is and write a brief description of the challenges in kerning. In particular, discuss the challenges for humans who perform the activity and the challenges in coding computer programs to perform the activity.

Find out what serif and sans serif fonts are in typography, and discuss the differences in terms of design and visual aspects, as well as in terms of computerisation of design.

For this part, submit

- an essay of at most 1,000 words.

[20%]

Part 2) Concrete exploration

For this part, you are to perform the activities below, and discuss what you find in a short report.

Look at the game you can find at `http://type.method.ac/`, which is about kerning. Work through the examples, and try to relate this to what you have read about kerning and written about in the previous part.

There are two brief *Processing* sketches, given at the end of this coursework assignment – that have been taken from `http://pages.uoregon.edu/park/Processing/process5.html` – which manipulate text. You can find these on the UoL VLE, called `text.pde` and `poem.pde`. Look at the code for each of them, and try to understand what it is doing. Run it, and play around with it. Think about the issues you looked at in the previous part while you do this, and if appropriate relate the font aspects to the kerning aspects.

You will notice that there are almost no comments in the code. You are required to make sure you understand how the code works, and to write appropriate comments into the code.

For this part, submit:

- your commented versions of each of `text.pde` and `poem.pde`, and
- your discussion of the insights that you've reached through using and experimenting with the three different software artefacts (the kerning game, plus the two *Processing* sketches).

[15%]

Part 3) Further developing a *Processing* sketch

Choose one of the two sketches that you looked at above, and develop it further, in a way that incorporates either concepts from the kerning ideas, or concepts from your discussions of serif and sans serif fonts. It is up to you how you do the further development – as long as it focuses on the typography aspects – but it is important that you justify and motivate any choices that you make in a short report that describes what you've done and why.

For this part, submit

- your further developed sketch, in a file called `part3.pde`, together with any other files that are necessary to run the sketch, and
- a brief discussion of the choices you made in further developing the sketch.

[15%]

Part 4) Creative artefact

For this part, you are required to develop a *Processing* sketch that uses text in an appropriate way, to make an artistic, creative or political comment. You should read Chapter 7 of the second volume of the subject guide, as well as the Creative Brief in the appendix of that volume, in order to obtain some guidelines as to how to approach the development of a creative artefact.

This is a somewhat open-ended question, that you can interpret in a variety of ways, and as such, there is no right or wrong answer. However, it is essential that you follow the brief given, which states that you must incorporate text into the sketch, and that it must make a comment as mentioned above.

Make sure that you include comments in your code, as well as using sensible variable names. Also think about how easy it would be to extend your code, as in general it is better to write flexible code where you can alter aspects of the sketch with ease; for example, using variables instead of hard-coded values where appropriate.

For this part, submit your sketch in a form that can be directly run in *Processing* by the examiners. This means you need to include all images, fonts, etc. that might be used by the sketch, in a folder that can simply be extracted and used. Please use sensible names for the files, and avoid underscores and hyphens, as some operating systems and *Processing* versions don't like these.

You are also to present a critical evaluation of your artefact, describing the motivation, the choices you made, and your assessment of its impact both artistically and technically.

[50%]

[TOTAL100%]

Submit the following by uploading to the VLE:

a. A single ZIP file containing two subfolders, named `artefacts` and `reports`.

b. The `artefacts` folder should contain two subfolders, that have the code required to run the extended sketch from part 3 of the coursework assignment, and the artefact developed in part 4 of this coursework assignment. Make sure for each of these sketches that you include any other files associated with the sketch (e.g. any image files or other multimedia that you might have used). You do *not* need to submit binary executable files.

c. The `reports` folder should contain one single pdf file, with the reports and essays from all four parts in it. Make sure that you give appropriate headings delimiting the work from each part, and start each one on a new page.

d. Your ZIP file should be named using the format *`FamilyName`* *`SRN`*`CO1112cw1.zip`

Important: Be very careful to check that your ZIP file contains all of your code and reports before you submit it. We strongly encourage you to try unzipping the file into a blank temporary directory on your own computer before you submit it, to check that it contains all of your `.pde` files, and that your `.pdf` file is also in the appropriate place.

It is essential that examiners are able to run your two sketches without having to make additional folders and move files around, so make sure that the file structure of your ZIP file is one that enables this.

Processing sketches for parts 2 and 3

text.pde

```
PFont font;
font = loadFont("Monospaced.italic-48.vlw");
textFont(font);

String mytext = "This is a quantity of text. You can put a lot of words here!";
String [] myArray = split (mytext, " ");

size(800, 600);
background(0);

for(int i=0; i<(myArray.length-1); i++){
  text(myArray[i], random(0,width), random(0,height));
  delay(150);
}
```

poem.pde

```
// "Dickinson Poem" by JOHN PARK
int i=0;
float rotater = 0;
String mytext = "A bird came down the walk: He did not know I saw;
He bit an angle-worm in halves And ate the fellow, raw.
And then he drank a dew From a convenient grass,
And then hopped sidewise to the wall To let a beetle pass.
He glanced with rapid eyes That hurried all abroad,--
They looked like frightened beads, I thought;
He stirred his velvet head Like one in danger; cautious,
I offered him a crumb, And he unrolled his feathers
And rowed him softer home Than oars divide the ocean,
Too silver for a seam, Or butterflies,
off banks of noon, Leap, splashless, as they swim.";

String [] myArray = split (mytext, " ");


void setup(){
size(800, 600);
background(0);
//frameRate(10);
PFont font;
font = loadFont("Monospaced.italic-48.vlw");
textFont(font);
println();
}

void draw(){
  rotater += .1;
  i++;
  if(i> (myArray.length-1)){
    i=0;
  }

  fill(255,45);  // white text, 25 alpha
  translate(mouseX, mouseY);
  rotate (rotater);
  text(myArray[i],0,0);
  delay(50); // in milliseconds
}
```

[END OF COURSEWORK ASSIGNMENT 1]

CO1112 Creative computing 1: Coursework assignment 2

Aims

The aims of Coursework assignment 2 are:

a. To develop your knowledge and practical experience of programming systems with motion.

b. To develop your knowledge and practical experience of programming systems with keyboard interaction.

c. To give you the opportunity to combine various techniques you have learned about during the course to produce a creative artefact.

Introduction

In Coursework assignment 2 you will be creating systems that involve moving objects. Before you start, you should have read Chapter 8 ('Motion') in Volume 1 of the subject guide. Later questions will require you to have read some sections of Volume 2 of the guide – we will point those out when we get to them.

An excellent supplementary text is the free online book *The Nature of Code*, by Daniel Schiffman (one of the lead developers of Processing). You can access the book at `http://natureofcode.com/book/`. Chapter 1 ('Vectors") and Chapter 2 ('Forces') provide in-depth coverage of the central concepts covered in this coursework assignment. We recommend you look at these chapters, whatever your level of existing knowledge in programming and mathematics.

A note on copying code from other sources: It is a normal aspect of programming to look at other people's code to get inspiration on how to solve a problem. This might extend to directly copying code obtained from elsewhere into your own program. If you do this, you should include a comment in your code to say which part has been copied, and where it came from. This is good practice whenever writing code, to acknowledge the original author. Note, however, that you will get *much* more out of this coursework assignment if you write all of the code yourself (after reading the appropriate sections of the subject guide and other sources), rather than copying code from elsewhere. That way, you can be sure that you really understand what the code is doing!

For each of the six parts below, do the following:

a. Name your Processing sketch `part`*`N`*`.pde`, where *N* is the part number (e.g. `part1.pde`, `part2.pde`, ...).

b. Include your name and SRN in a comment at the top of the sketch.

c. Include an appropriate number of comments in your sketch to explain what it is doing.

d. Write a brief description for each part, describing what (if any) problems you encountered, and how you tried to solve them (whether or not successfully). If there are any aspects of your sketch that are unusual, or that you are particularly proud of, or that you think could be improved upon, write about those too. For Parts 1–5, a single paragraph for each part should be sufficient. Your written description for Part 6, including the requested critical evaluation, should be a little longer (about one side of a standard A4 page of text would be appropriate).

Part 1) Basic motion

Create a Processing sketch of a ball moving across the screen, with the following properties:

- Screen size 600 by 600 pixels
- Frame rate: 60 frames per second
- Background colour blue
- Ball colour white, diameter 30
- Ball start position: top-left corner of screen
- Ball velocity: 2 pixels per update in x direction, 3 pixels per update in y direction
- (There is no need to detect when the ball runs off the edge of the screen for this part.)

Name your sketch `part1.pde`.

When you write your code, think about how easy it will be for other people (including the examiners!) to read and understand what you have written. Adding comments and sensible variable names can very much improve the readability of your code. Also think about how easy it might be to extend your code – for example, using variables instead of hard-coded numbers can make it much easier to change aspects of the sketch's behaviour, as well as making the code more readable.

[20%]

Part 2) Bouncing at screen boundaries

Make a copy of your sketch from Part 1 and save it as `part2.pde`.

Add code to your sketch to detect when the ball reaches a screen boundary (i.e. the top, bottom, left and right edges of the screen). When it does, make the ball bounce off the boundary by reversing the appropriate component of the ball's velocity (i.e. when the ball bounces off the left or right boundary, reverse the `x`-component of velocity, and for the top and bottom boundaries reverse the `y`-component).

For full marks, think carefully about exactly when the ball should bounce at each of the four boundaries.

[10%]

Part 3) Gravity

Make a copy of your sketch from Part 2 and save it as `part3.pde`.

Add gravity to your sketch, so that the ball accelerates towards the bottom of the screen (and, of course, bounces when it hits the bottom, which we have already implemented).

Experiment with different magnitudes of gravity, and choose one which produces a visually pleasing effect.

[10%]

Part 4) Interaction with other objects

Make a copy of your sketch from Part 3 and save it as `part4.pde`.

Add a new rectangular block to your sketch, with width 150 pixels, height 30 pixels, and position of top-left corner at (300,400). Choose a suitable colour for the block.

Add new code to your sketch so that the ball bounces off the block when it collides with it from any direction.

[10%]

Part 5) User interaction

Make a copy of your sketch from Part 4 and save it as `part5.pde`.

In this part, we are going to add some simple user interaction to our sketch. If you haven't already done so, read Section 3.4 of Volume 2 of the subject guide, to understand how to detect when the user presses keys on the keyboard. Also look at example sketches in the Processing example library and on the Processing website for further examples.

Modify your code so that the user can move the block leftwards by pressing the 'z' key on the keyboard, and rightwards by pressing the 'x' key.

Think about how quickly the block should move when a key is pressed. Choose a speed that gives a pleasing feel, not too fast and not too slow. Also think about what you should do when the block reaches the left or right edges of the screen. Finally, try to modify the code so that the ball still bounces off the block appropriately, even after it has been moved.

[10%]

Part 6) A Rio 2016 Olympics-themed creative artefact

In the final part of Coursework assignment 2, we are going to create something with an Olympics theme, to celebrate the upcoming Rio 2016 Olympic games.

Create an Olympics-themed sketch that in some way incorporates the motion and collision-detection techniques that we have used in Parts 1–5. Also feel free to incorporate any other techniques that you have learned about so far in *Creative computing 1*. Name the sketch `part6.pde`.

This is a very open exercise and you may interpret these instructions any way you wish. We are looking for originality, creativity, technical skill and the overall effectiveness of your final sketch.

To give just a few examples, you might think about creating a simple game to represent one of the Olympic sports, or you could think about the Olympic logo, the stadia and spectators, the opening ceremony, etc. You could think about doing something in 3D rather than 2D, using texture mapping, etc. These are all just random suggestions – feel free to do whatever you like.

Write a short critical evaluation of your sketch, describing the motivation, the choices you made, and your assessment of its impact both artistically and technically.

[40%]

[TOTAL 100%]

Submit the following by uploading to the VLE:

a. A single ZIP file containing two subfolders, named `code` and `reports`.

- The `reports` folder should contain a single PDF document containing your written descriptions of each of the six parts.
- The `code` folder should contain six subfolders, named `part1`, `part2`, etc. Each subfolder should contain your Processing sketch for that part, named as specified above (e.g. `part1.pde`, `part2.pde`, etc.) and any other files associated with the sketch (e.g. any image files or other multimedia that you might have used). You do *not* need to submit binary executable files.
- Your ZIP file should be named according to the following format:
  `FamilyName``_`*`SRN`*`_CO1112cw2.zip` (e.g. `Smith_150123456_CO1112cw2.zip`)

Important: Be very careful to check that your ZIP file contains all of your code before you submit it. We strongly encourage you to try unzipping the file into a blank temporary directory on your own computer before you submit it, to check that it contains all of your `.pde` files.

[END OF COURSEWORK ASSIGNMENT 2]