

CGRA151 2019 Assignment 1: introduction to Processing

Learning objectives

Core assignment: be able to get Processing working, be able to write a simple program to draw a range of static Processing primitive objects, be able to submit Processing sketches (programs) as a zip file in the School submission system.

Completion assignment: be able to handle both random and interactive elements in Processing sketches and be able to generate multiple objects.

Extension assignment: be able to apply what you have learnt to produce an original piece of “digital art”.

Practical matters — how to submit your work.

You will write up to five Processing sketches (“sketch”=“program”): one for core, three for completion, and one for extension. Each sketch will be in its own sub-directory in your Processing sketch directory. To submit, you need to use the zip utility to pack all the directories into a single zip file. You then upload that one zip file to the ECS submission system. For example, if you complete all five sketches, and give them the names you are told to give them, then you would run this command to make the zip file:

```
zip -r A1submission.zip A1Core A1CompA A1CompB A1CompC A1Extn
```

The “-r” option tells zip to include all of the files inside the directories. The file that you submit is `A1submission.zip`. See the course website for details of how to use the submission system.

Your sketches must run on the version of Processing used on the ECS machines without needing to be modified.

Marking

Marking will be done by tutors in one-on-one sessions with you during your usual Lab time. For this assignment the marks are divided up this way:

Core assignment: completing this satisfactorily (and demonstrating that you can zip and upload your sketches) can provide up to **55%** of the marks.

Completion assignment: completing this will provide a further **25%** of the marks.

Extension assignment: completing this will provide a further **10%** of marks.

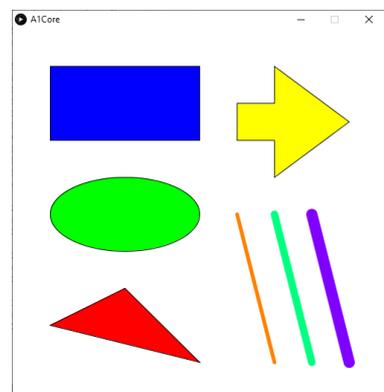
Code quality: the final **10%** of the marks are allocated to the quality of your code.

Core assignment — getting Processing working

Write a Processing sketch that reproduces (approximately, not exactly) the image to the right. It requires you to understand how to use the following Processing functions:

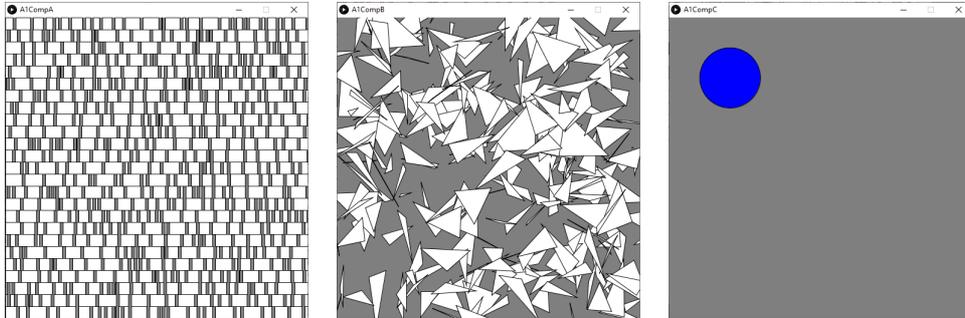
```
size(), background(), fill(), stroke(),  
strokeWeight(), rect(), ellipse(),  
triangle(), line(),  
beginShape(), vertex(), endShape()
```

Save this sketch as A1Core.



Completion assignment — patterns and interaction

You are required to write three separate Processing sketches that reproduce, using the `random()` function where necessary, patterns that look similar to the following three images. The first two sketches are static: they do not change once drawn. The third sketch is interactive: it responds to mouse movements and mouse presses.



The third sketch requires use of the `setup()` and `draw()` functions. The `draw()` function should draw a circle centred on the current mouse position. The circle should be red if a mouse button is pressed and blue if it is not pressed. Use the built-in variables `mouseX`, `mouseY`, and `mousePressed`.

Save these sketches as `A1CompA`, `A1CompB`, and `A1CompC` respectively.

Extension assignment — artistic expression

Processing has a range of geometric shapes: rectangles, triangles, ellipses. Demonstrate that you have understood the basics of Processing by picking one type of shape (e.g., rectangles) and using groups of that shape to produce either an interesting static image, a pleasing animated sequence, or an interactive piece of “art”. The aim here is that you use your imagination to create something that looks interesting to you. It might be based on some mathematics, it might be based on random number generation, or it might be based on some other idea of your own. The challenge is that you are allowed to use only one type of shape. This is to help you focus. You need to be able to explain to the marker how you implemented it and also the underlying inspiration.

Save this sketch as `A1Extn`.

Igor Stravinsky said *“The more constraints one imposes, the more one frees one’s self. And the arbitrariness of the constraint serves only to obtain precision of execution.”*