Computer Science D01
University of Toronto Scarborough

Project Deliverable #2

---

1 Getting to know the code base

In this part of the project you will familiarize yourself with the existing codebase. As you know, we will be working on http://matplotlib.org/.

- Check it out of the repository, configure, and run.
- Read through the documentation, learn to use it in a Python program.
- Investigate available tools that will help you with the assignment. Among others, you will need a UML drawing tool (simple lightweight tools will work fine for this), and a reverse engineering tool.
- Browse through the repository, familiarize yourself with the structure.
- Use a reverse engineering tool to generate a UML class diagram representing the classes, subclass relationships, and associations in the source code. You will need to edit the generated model to capture information missed by the tool, to remove unnecessary detail, etc.
- Draw a higher-level diagram to show the overall architecture of the system. Use any appropriate UML notation (e.g. packages, components, interfaces, etc). Be sure to show clearly where the classes belong in this architecture, and what external packages the system interacts with. As always, the exact UML notation that you use is much less important than the modelling decisions you make: what is important enough to include, what to omit, and how to structure your diagrams.
- Identify three different design patterns used in the system, and show how each is implemented (using UML diagrams). Be sure to illustrate the pattern with both structural views (e.g. class or object diagrams) and behavioural views (sequence diagrams). Your TA will examine the piece of code you refer to, in order to make sure you correctly identified the design pattern.

2 What to submit?

- A commentary on the architecture of the system, highlighting any interesting aspects of the design (e.g. architectural style, degree of coupling, etc), discussing the quality of the architecture used in the system, and suggesting possible improvements where appropriate. Use UML diagrams to illustrate the points you wish to make.
- A description of each of the design patterns you identified, along with both structural and behavioural UML models, and links to the corresponding code.
- Any other UML models you generated, as appropriate. Be sure to include at least enough views so that you show how everything fits into the overall system structure.

2.1 Marking

- Overall architecture
  - It is clear to the reader that the team correctly understands the architecture of matplotlib. (5 marks)
  - A reader, initially unfamiliar with the architecture of matplotlib, has a good overview of matplotlib after reading the report. (5 marks)
  - UML diagrams are correct, clear, and helpful. (5 marks)
- Design Patterns:
  - Each pattern:
    - The pattern is actually present in the source code (3 marks)
    - The diagrams are correct, clear, and helpful (2 marks)
- Presentation and Quality of Writing (10 marks)

  It is up to you to choose the format of all write-ups in the course: you can produce either `pdf` documents or collections of `html` files. Whatever format you choose, make sure it looks professional and is very easy to read: the TAs will not have much time and you need to convince them you did an excellent job!

  The following will always be considered when grading your work in this course:
  - Presentation: the report is well formatted, easy to read, and easy to navigate.
  - Quality of writing: language, grammar, clarity, professionalism.