Computer Science D01
University of Toronto Scarborough

Project Deliverable #3

---

1 What you will do

For this deliverable, you will select at least two issues (bugs and/or features), complete the implementation of them, and provide test cases suitable to demonstrate that the changes have been correctly implemented. Note that it is possible, for this phase of the project, that a `matplotlib` committer may be simultaneously working on the same issue. If this happens, please resist the urge to look at the code they commit and focus only on your team's implementation.

This part of the project requires you to use your judgment about which change requests to select for implementation. There is no "correct" choice: you will be given credit for selecting changes that can be implemented correctly in the time available, and which are most likely to satisfy the users. Note that users are more likely to value simple fixes that work reliably over more ambitions features that are incomplete.

2 What to submit/present and when

There are two steps in this deliverable.

Step 1 Examine the `matplotlib` issue list in GitHub. From the issue list, select a handful (a minimum of five) of promising bugs or features to examine further. The list of bugs you have selected for investigation should be in your repo by **Monday, Feb 27, at 10:00 a.m.** Reproduce the bugs and demonstrate them to the TA during the meeting that week.

Select a minimum of two items from your shortlist to implement and test. It may involve estimating the effort required to implement each change, and identifying any anticipated risks. It should involve determining which team members are allocated to work on which task and when. During the meeting with your TA, be prepared to explain your choices and have some ideas on how you will try to fix the bugs.

Step 2 Implement your selected items. Make sure all the changes are in your repository by the due date — your TA will clone the code from your repository to run it, as part of marking the deliverable. Make sure the new/modified code is well-documented!

Document your bug fixes and write test cases to demonstrate that the changes have been implemented correctly. Design these as "customer acceptance" tests: i.e. a description of the steps a user needs to carry out to check that the items requested.

Submit a set of tests, described in a form that would allow any user to download your new version of a `matplotlib` component, run it, execute the tests, and determine that the software works correctly. You can assume that the user has a basic working knowledge of Python and `matplotlib`.

Write a brief technical commentary on how your changes affect the design and/or the code of `matplotlib`. List all relevant `matplotlib` source code files that were added, modified, or removed as part of your implementation work.

2.1 Marking

- Step 1 (5 marks)
  - Demonstration of at least 5 bugs.
  - Understanding where each of these bugs is in the code, more or less, or at least understanding where one would start investigating.
- Step 2 (30 marks)
  - At least two bugs are fixed (correctness). (10 marks)
  - Quality and documentation of changes/fixes: no hacks! (5 marks)
  - A well-designed and well-explained acceptance test suite. (5 marks)
  - Clear evidence of a well-structured development process, good use of the repo and of the selected project management tools, etc. (10 marks)
- Presentation and Quality of Writing (5 marks)
  Same guidelines and expectations as before.