MATPLOTLIB: PROJECT DELIVERABLE 4

Team 16 (Do It Tomorrow)

CSCD01

March 17th, 2017

Team Members:
Mengzhe Lu
Jubin Patel
Derek Chow
Daniel Karas
Gauravjeet Kala

Table of Contents

Selection of Issues

As a team, we have discussed and decided to look into the following issues found on the issue list of matplotlib:

GitHub Issue Link	Description
Issue 3293	In the current implementation of matplotlib, default markers for line plots can be set through rcParams['lines.marker'], however there is no way to set this default marker's face color. Therefore, a new key should be added to the rcParams object to include the default marker's face color.
Issue 8057	When plotting a graph, it is not currently possible for the user to choose the markevery value using numpy integers. It currently only works when the markevery value is using built-in integer types.

Selected Issue to Implement:

As a team, we decided to select issue 3293 as our issue to implement, over our second issue, issue 8057. Since we find that although issue 8057 is clearly seen as an issue, there is currently a workaround to be able to plot with markevery=3. This issue can be mainly seen as that we cannot use a numpy integer for this field in a plot.

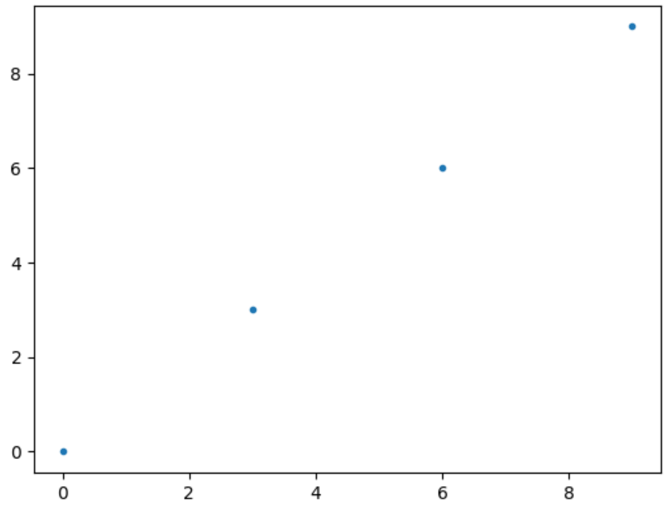
Another reason we chose issue 3293 is that issue 8057 can be resolved by simply adding several explicit type checking, which we find is not a substantial new development to a project like matplotlib. On the other hand, issue 3293 seems more challenging as we are trying to completely add a new parameter to rcParams. In this case, we will need to add to the template rcParam file, as well as find every place that uses linestyle and check this new rcParam. Therefore, we will need to spend our resources not only coding the resolution to the issue, but also allocate time to understand which parts of code is using rcParams, and check for this newly implemented parameter.

Issue 8057: Value Error ‘markevery’

Description:

In issue 8057 of matplotlib, when creating a plot with the parameter ‘markevery’, it must be required that the value provided for the parameter is a built-in type or else matplotlib will produce a **Value Error**.

Therefore if the user were to use an integer value, then the code would run with no issues. For example if the user uses the following code: *plot(range(10), ".", markevery=3)*
It will produce the following plot:



However when a numpy integer is introduced and replacing the “3” in the markevery parameter such as: plot(range(10), ".", markevery=np.int_(3).

matplotlib will produce the following error:

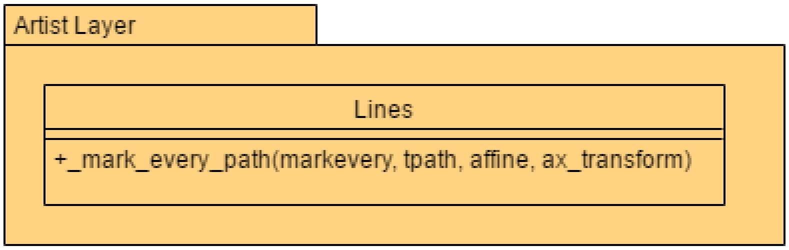
> *“ValueError: Value of markevery is not recognized; markevery=3”*

Implementation Design:

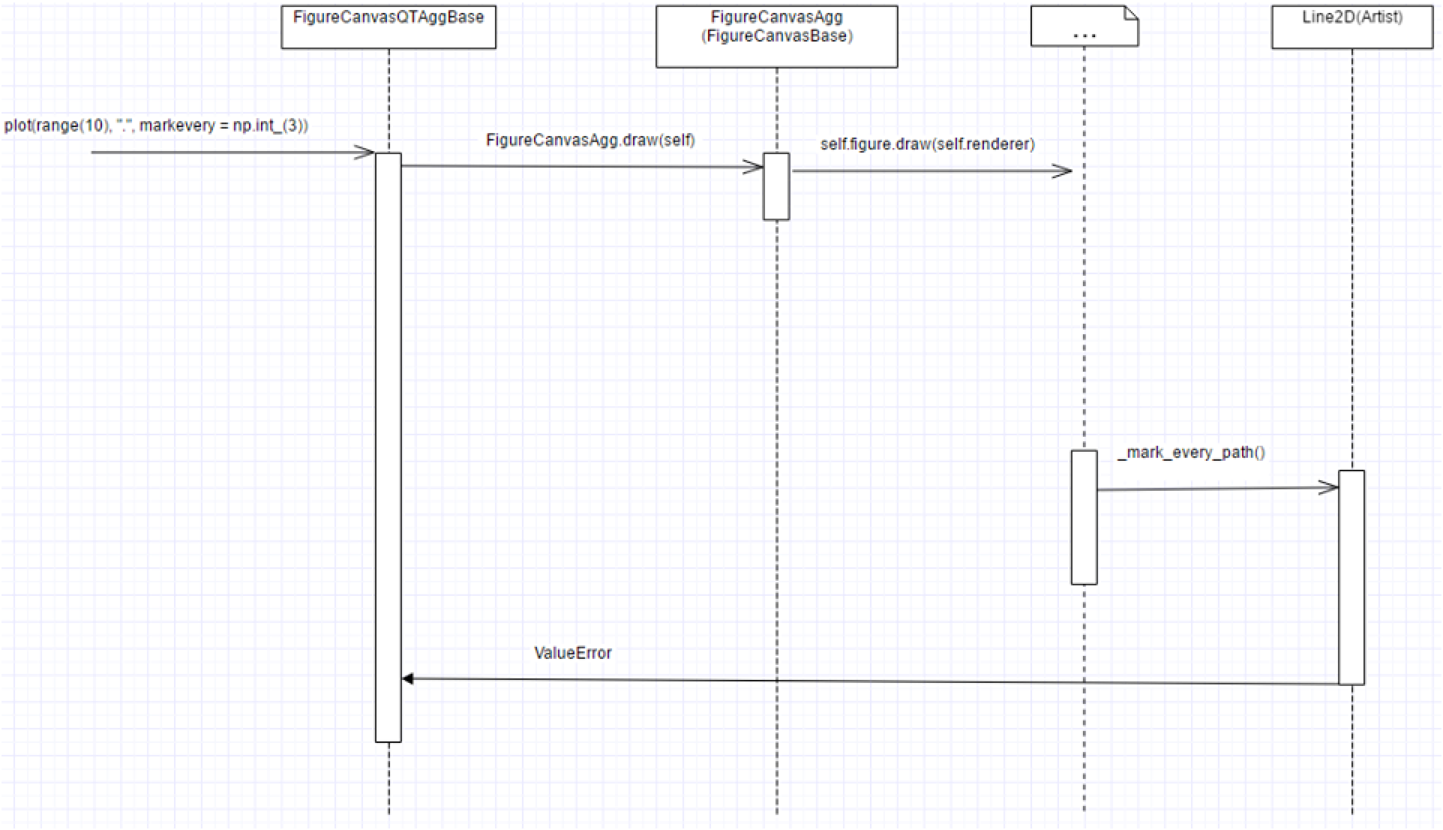
It seems that the method _mark_every_path in lines.py will need to be modified in order to fix this issue. Specifically the checking to see if numpy integers are being passed in and how to handle this case if they are.

The implementation of this bug fix has to do a lot with modifying existing code base rather than implementing brand new methods to fix this issue. Therefore we don’t see the changes impacting other code in matplotlib. Furthermore, as we implement this fix, we also don’t expect existing matplotlib functionalities to be impacted either, due to the fix being contained in one method of one file.

UML Diagram:



Sequence Diagram:


FigureCanvasQTAggBase
FigureCanvasAgg
(FigureCanvasBase)
...
Line2D(Artist)
plot(range(10), ".", markevery = np.int_(3))
FigureCanvasAgg.draw(self)
self.figure.draw(self.renderer)
_mark_every_path()
ValueError

Issue 3293: Default Marker Face Color

Description:

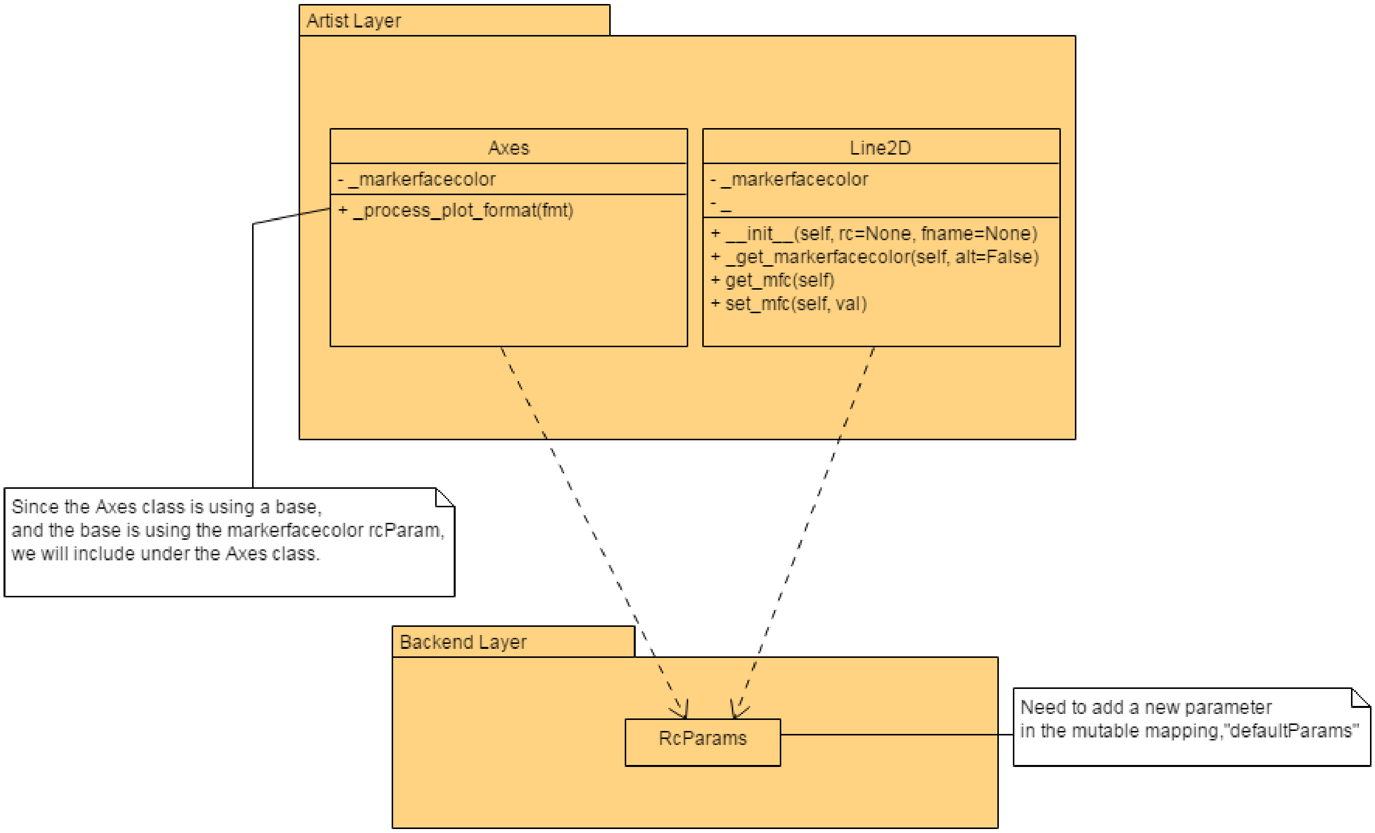
In issue 3293 of matplotlib, when creating a line plot, there is currently no way of setting the default marker's face color. In this case, there does not exist a parameter under the rcParams object such as 'lines.markerfacecolor'. Therefore for this issue to be resolved, we will need to include a 'lines.markerfacecolor' key into our rcParams object, and potentially store the selected default color as the value of this newly created key. However, to fully implement this feature, we will need to adjust and allocate checks to the original rcParams: '*linestyle'* to check this newly added rcParams properly.

Implementation Design:

For the implementation of this feature we will be specifically storing the default marker's face color into the rcParams object. In this case we will be modifying the rcsetup.py file, specifically the mapping, '*defaultParams*' as well as adjusting all places that uses the '*linestyle*' parameter to check for this newly added parameter. Specifically since this is a new rcParam, no current code uses this as a default. Therefore, we will need to search for every section of code that is able to control the marker face color, and add logic to check that the user did not explicitly pass in a value.

The implementation of this bug fix has to do a lot with modifying existing code base rather than implementing brand new methods to fix this issue. However, this fix does tie into other current functionalities of matplotlib. Therefore, many of the functionalities, specifically the ones that controls the default marker's face color, will be impacted by this change.

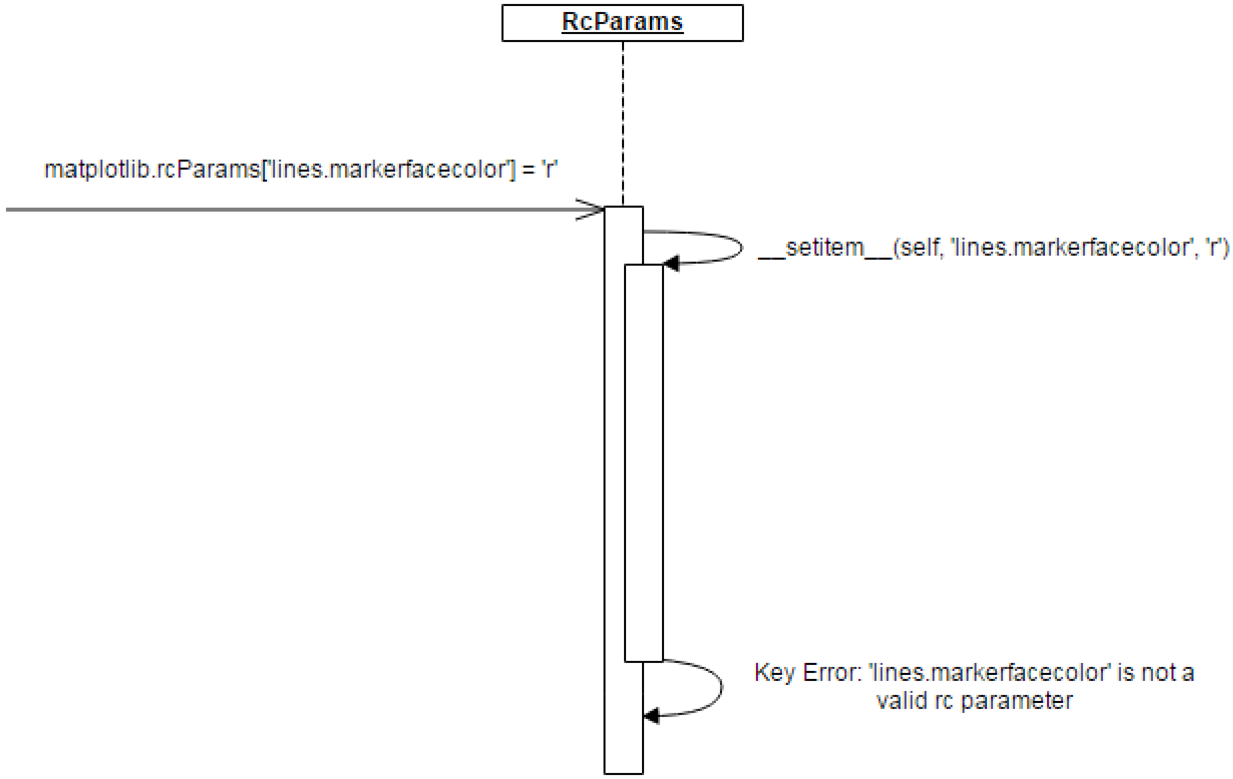
UML Diagram:



Since the Axes class is using a base, and the base is using the markerfacecolor rcParam, we will include under the Axes class.

Need to add a new parameter in the mutable mapping,"defaultParams"

Sequence Diagram:


RcParams
matplotlib.rcParams['lines.markerfacecolor'] = 'r'
__setitem__(self, 'lines.markerfacecolor', 'r')

Key Error: 'lines.markerfacecolor' is not a valid rc parameter

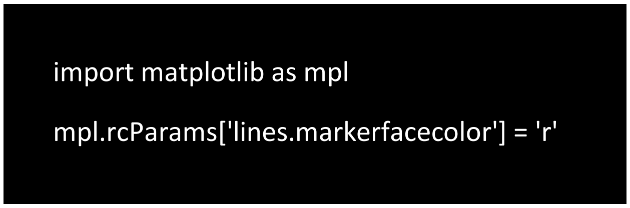
Testing

Acceptance Testing:

Currently matplotlib doesn't allow the user to set the default marker face color, and when the user tries to set said parameter, they are encountered with the following error:

Key Error: 'lines.markerfacecolor is not a valid rc parameter.
See rcParams.keys() for a list of valid parameters.

To recreate the error, the user can type the following python code, assuming they have matplotlib and its necessary dependencies installed.



```
import matplotlib as mpl

mpl.rcParams['lines.markerfacecolor'] = 'r'
```

Furthermore to see if the colours are changing, you can edit the second line where you have 'r' which is indication for the colour red or 'b' which is for blue.

Alternatively you can also check if the option to set the default marker face color is available through the following method as well:

1 - Run the following script:

```
from matplotlib import rcParams

print(rcParams['lines.markerfacecolor'])
```

2 - It should not give a Key Error or crash. Rather, it will print out the default marker face color (as declared it would be declared in the line properties in lib/matplotlib/rcsetup.py:913).

Unit Testing:

If the fix has been implemented, you can run the following unit test suite and make sure it passes all tests to ensure the feature was implemented, as well as running the previous code that raised the error. The unit test suite can be found in the file "test_issue3293.py" under the "Deliverable 4" folder.