Student: Zhe(Kevin) Chen

Student number: V00819544

Date: Feb 03, 2018

Toyplot

Table of Contents

Assignment Overview

For target project, pick a quality attribute, create a quality attribute scenario, and document how the project source code realizes this scenario. Provide discussion on how the project can handle changes in this scenario in the future.

Introduction

Toyplot is an open-source python library from Sandia National Laboratories that develops beautiful interactive, animated plots that embrace the unique capabilities of electronic publishing and support reproducibility. It creates the possible data graphics out-of-the-box, maximizing data ink and minimizing chartjunk. It also comes with minimalist interface for engineers and scientists to explode.

The following content will be discussing one use case scenario and one growth scenario with their investigations based on one chosen quality attribute.

Quality Attribute Scenarios

Use Case Scenario

In the section I choose quality attribute Performance and create a simple scenario for Toyplot as it is basically a plotting toolkit, the most common use case is related to how it performs.

When a task calls a function in library to generate a normal sized plot, the whole process time must take less than 0.5 seconds from called to plot result.

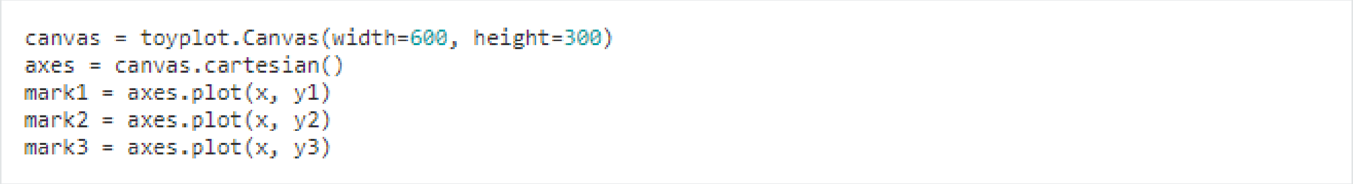
Aspect	Details
Scenario Name	Low latency on task performance

Aspect	Details
Business Goals	Efficient process speed to ensure user having low latency experience
Quality Attributes	Performance
Stimulus	Need for generating a plot with 3 sinusoids from user
Stimulus Source	User and function in that module being called
Response	Display the generated plot
Response Measure	< 0.5s process and render capability

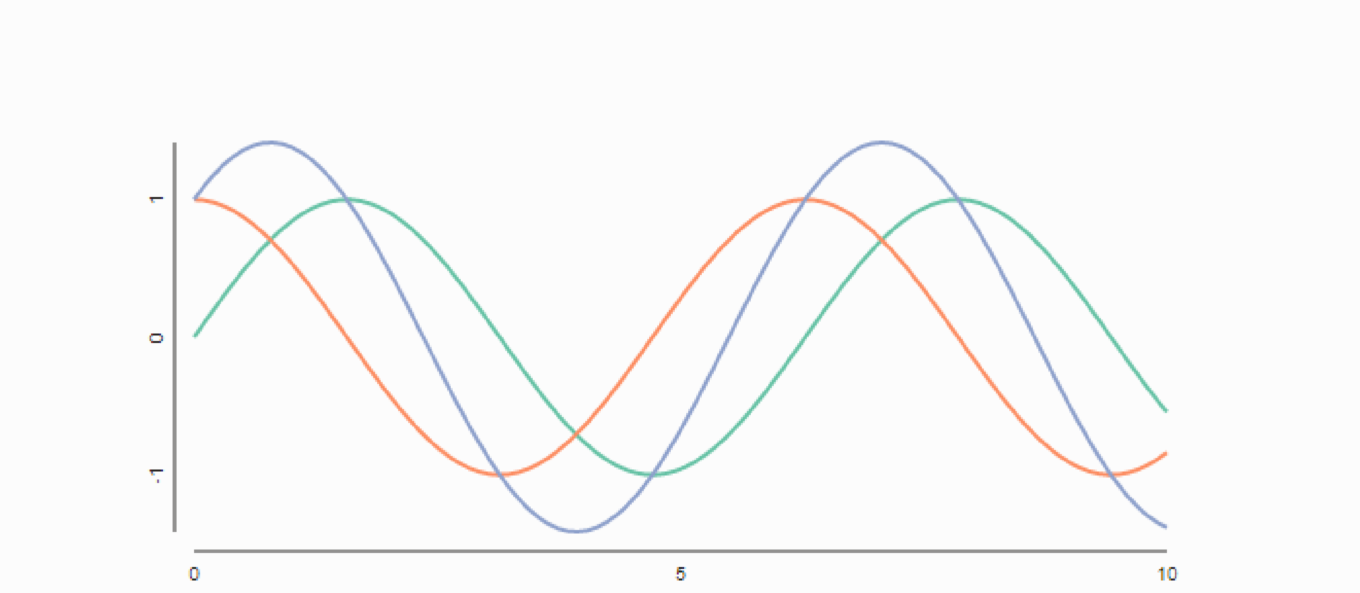
My response measure seems to be a reasonable guess since it is hard to get an estimation without getting hands on the toolkit and dig deeper into the source code.

Use Case Scenario Investigation

In order to generate a basic three-sinusoids-plot as introduced in The Toyplot Tutorial using the toolkit, it requires these steps to get the following result:



```
canvas = toyplot.Canvas(width=600, height=300)
axes = canvas.cartesian()
mark1 = axes.plot(x, y1)
mark2 = axes.plot(x, y2)
mark3 = axes.plot(x, y3)
```



In the scenario above, my main focus will be on canvas file because it essentially creates a Canvas class and calls cartesian function.

When initialize the class, it only assigns basic width and height for the canvas, therefore it is an instant event. After having the canvas of the plot, it uses the cartesian function to add a set of Cartesian axes to the canvas which is also an instant event because it creates a Cartesian class from coordinates which looks alike to Canvas class. Now that I can see the only part that could be time consuming is to add different sinusoids to the plot, and since the scenario only takes three marks, thus it will not take more than 0.5 seconds to process and render the plot.

Growth Scenario

Now we consider increase the performance of the toolkit by increasing sinusoid number in the plot from 3 to 5000 and analyze the performance to see if the low latency requirement can be satisfied or not.

Aspect	Details
Scenario Name	Low latency on task performance
Business Goals	Efficient process speed to ensure user having low latency experience
Quality Attributes	Performance
Stimulus	Need for generating a plot with 5000 sinusoids from user
Stimulus Source	User and function in that module being called
Response	Display the generated plot
Response Measure	< 0.5s process and render capability

Growth Scenario Investigation

It mainly relies on its data handling ability to add 5000 data into to the plot. From the inspection of the source code, Cartesian.plot function is in charge to plot data into the canvas once upon each call. In that function it contains essentially mapping of its attributes and thus adding more sinusoid seems to be at a linear growth rate. Therefore for a task such that generates a plot with 5000 sinusoid might be able to have processing and rendering time lower than 0.5 seconds, but as the expansion of the data required, the more processing and rendering time is also required and eventually exceed the response measure.