py5 Documentation and Additions

A GSOC 2022 proposal

Mentor: @tabreturn @hx2A

Name: Tushar Gupta
Email: codingid6@gmail.com
Telephone: 91-8052717667
Github: tushar5526
LinkedIn: tushar55
Portfolio: https://tushar5526.github.io/

Introduction

"py5 is a new version of Processing for Python 3.8+. It makes the Java Processing jars available to the CPython interpreter using JPype. It can do just about everything Processing can do, except with Python instead of Java code."
– py5.ixora.io

As the library is in its initial stage of development, it requires thorough tutorials and accompanying examples to get beginners (and, secondarily, more advanced users) familiar with py5 basics, such as drawing using py5 commands, general Python concepts and best practices; then progress in a manageable way to more advanced py5 skills like 2D transformations, PShape, Curves and Python techniques, like OOP, working with Py5Vector, list comprehensions etc. These Tutorials and examples will help deliver the best of both Python and Processing in a manner useful for both beginner and experienced developers.

The documentation process would also aid in improving py5 providing feedback to its developers to indicate if the feature or implementation of the Processing API should work as it does the Python port. Moreover, reveal if there is a different approach that might improve things or be more intuitive for users by prompting discussions in the community.

Base Goals

- **Getting Started Page**
  - **Install py5**
    - The Getting Started page is already well written, it focuses on installation using conda which is the best method in my opinion as well. We could also add detailed steps for Linux users who want to install it without using conda (for users who just want to use processing and are not looking for the data science/ML benefits of the conda)

    - Mentioning the pros of the py5 Thonny plugin as well.

    - In Simple Steps, we should mention that on installing java-17 through `install-jdk,` we also have to update **JAVA_HOME** variable manually if that variable is already pre-configured on there is a java version present on your system.
      This step is documented but only in the detailed steps and it would be better to mention it in *Simple Steps* as well. Actually, I faced this issue while setting up the library on my local system.

    - Add a “What next page” at the end of the Getting Started page which would direct the user to the **Tutorials & Examples page** or a **Structured approach to py5** as discussed next.

- Tutorials and Examples & Beginners’ Guide to Processing + Python

  - **Text Tutorials and Examples of Processing**
    - Each Processing port has a set of great tutorials and examples written by professionals who are great at their craft and this is the reason we see them being adapted to different ports of Processing.

    - So, a completed adaptation of Processing text tutorials and examples is mandatory as users from different ports of Processing will find it convenient to see a Python adaptation of a tutorial/example that they followed in some other Processing port.

    - Examples and tutorials would not just be a direct translation from Processing et.al because py5 and Python offer better ways to do things. I will be focussing to use vectorized and efficient code, writing examples in the “Pythonic” way (like list comprehensions instead of standard for loops, using np_pixels instead of pixels)

    - The current documentation system (jupyter notebook) is great and flexible for the project.

    - py5 static mode provides us with the flexibility for users to run a few examples like p5.js

  - **A new structured pathway to learning the ideas of processing and by extension Python as well as its cons + how to do stuff in a Pythonic way**

    - Processing might seem overwhelming for complete beginners, but a structured approach to Processing will help them get started easily.

        - Regarding this reply of @tabreturn, it's useful if the tutorial content follows a progression -- from getting started to intermediate -- so that the user might read the tutorial content in sequence to learn programming py5 (and, by extension, Python) from scratch. Some more accomplished person, however, might jump to a specific section.

        - The way I think it would be possible is through the use of **Examples** from p5.js.

        - We provide a structured path for new users to visit the **examples and text tutorials** in the manner we recommend.

        - I like how examples are done in p5.js, we could do something similar for py5 and the flow would be redirecting users to these examples to get started without getting overwhelmed.

        - Thanks to @tabreturn and @villares suggestions in the discussion. I am preferring to use the flow provided by them due to their experience teaching Processing and the flow is great as well to follow.

Flow to follow in structured path to py5 for beginners

- Installation and getting py5 running
- Drawing 2D primitives (the basic coordinate system, colour, fills, strokes).
  Note: I use static mode until point 9.
- Variables (and dealing with different data types -- for creating visual output, of course)
- Drawing more complicated shapes (curves, vertices, contour functions)
- Working with text (basics of working with strings and text functions)
- Conditional statements (if, else, elif to produce visual results)
- Iteration (while and for loops to produce visual results; break, continue)
- Randomness (random functions for making tiled patterns)
- Motion / Animated sketches (the setup() and draw() functions; frame... functions)
- Transformation (transformation functions, push/pop matrix)
- Working with lists (standard Python list stuff with visual results)

- Reading data (CSV files to generate charts)
- Dictionaries (standard Python dictionary stuff, and maybe loading JSON)
- Functions (defining function, args, return, etc. to produce visual results)
- Trigonometry essentials (using trig functions for animated results)
- Object-Oriented Programming (standard Python OOP stuff with visual results)
- PVector (combined with OOP to produce visual, animated results)
- Mouse and Keyboard Interaction

        - p5.js Examples cover a wide range of topics we are looking for, but if something is missing specific to Python like integrating some libraries.

The different py5 modes:

Support for jupyter notebooks, Cairo for high-res rendering & many other features were highly demanded in p5py and there are different issues mentioning them.

py5 has already tackled all these issues.

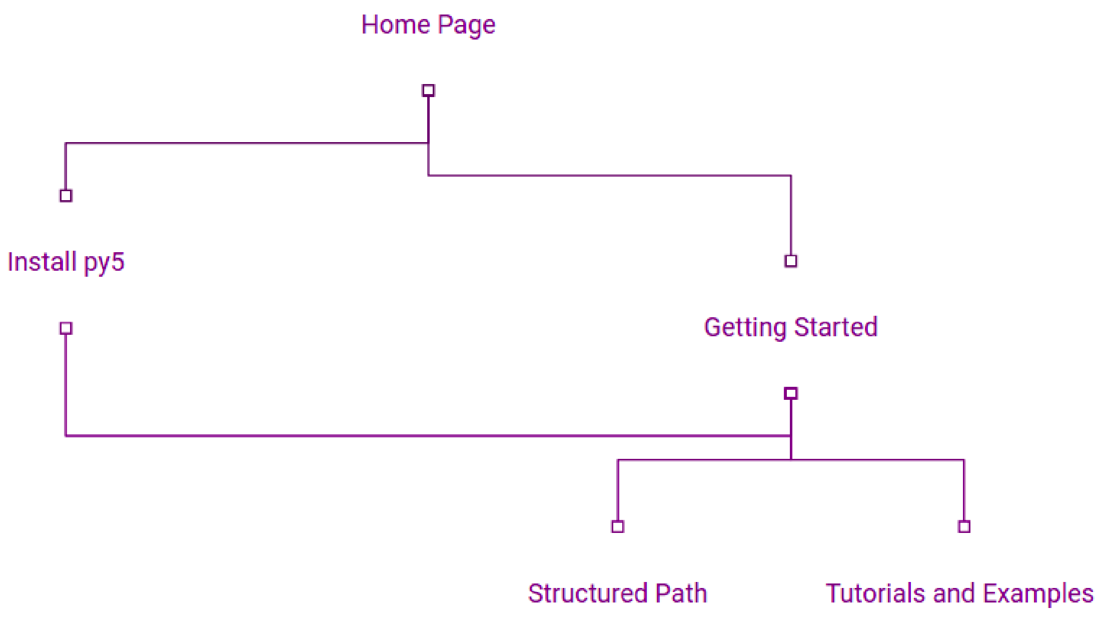
I will be trying to leverage those different modes to demonstrate examples. Static mode is great to run the examples in a browser which lets users quickly try out some changes similar to what p5.js has accomplished.

But not all examples (interactive examples) are possible through the static mode, so we could use a mix of different py5 modes to demonstrate different examples.

We can only display static sketches using static mode in py5 with the help of Thebe or Binder.

For interactive sketches and non-static sketches, p5.js could be used to display the sketches alongside python code for that sketch.

Alternatives such as http://berinhard.github.io/pyp5js could not be used because pyp5js is built on top of p5.js and uses p5.js naming and coding conventions, which is a lot different from py5's convention.



The general flow for new users visiting py5

Timeline

The project will play an important part in shaping the py5 community. I think it would fit just fine as a medium-sized project ~175 hours or a 12-week long program, but I would be able to extend it to a bigger timeline as well if required.

- **Community Bonding Period (May 20 – June 12)**
    - Interacting with mentors
    - Trying out different examples locally and fiddle around with different py5 modes
    - Answering questions related to py5 on Discourse, going over different issues currently open for the project.
    - Get more concrete feedback on my thought process and the approach I will be beginning with to start the project.
    - Get familiar with the current documentation system

- **Week 1 (June 13 - June 20)**
    - Implement the additions for Getting Started Page.
    - Provide necessary links wherever required
    - Add the “Where to next page”

- **Week 2 - Week 3 (June 21 - July 4)**
    - Start with the Tutorials Page
    - Cover the first six Beginner tutorials.
        - Getting Started by Casey Reas and Ben Fry
        - Processing Overview by Ben Fry and Casey Reas
        - Coordinate System and Shapes by Daniel Shiffman
        - Color by Daniel Shiffman
        - Interactivity by Casey Reas and Ben Fr
        - Objects by Daniel Shiffman

- **Week 4 - Week 5 (July 5 - July 19)**
    - Work with the intermediate tutorials
        - Two-Dimensional Lists by Dan Shiffman
        - Images and Pixels by Daniel Shiffman
        - Strings and Drawing Text by Daniel Shiffman
        - 2D Transformations by J David Eisenberg

- **Week 6 - Week 7 (July 20 - August 3)**
    - Move on to advanced tutorials
        - P3D by Daniel Shiffman
        - Anatomy of a Program by J David Eisenberg
        - Vectors by Daniel Shiffman: An introduction to using the PVector class in Processing.

- **Week 8 - Week 10 (August 4 - August 18)**
    - Implement the Examples section https://p5js.org/examples/ for py5
    - Start working on the "Recommended path to py5" page.

- **Week 11 - Week 12 (August 19 - September 5)**
    - Continue with the *"Recommended path to py5"* page.
    - Add a few Python-specific examples to cover the Pythonic way of coding.
    - Add py5 specific tutorials on **py5 Magics and py5 Functions.**

Students under GSoC have to work 175 hours during the 12 weeks of the program, therefore I will be devoting my 3+ hours on weekdays and 4+ hours during the weekends as per my schedule.

I will be having my semester examinations at May end and will not be able to contribute much on the exam days, but will be able to contribute to the project in preparation gaps provided during the exams.

I will be completely free after my end sem exams and would cover the backlogs if any due to examinations.

Apart from that, I am available during the summer for GSoC.

Extended Goals (After September)

- Help set up tests for py5 and develop a CI-CD pipeline for effective development
- Base goals would get me acquainted with a lot of parts of the codebase and have a great and deep understanding of Processing that I could put to use in setting up tests
- I have experience setting up visual tests and working with GHA, and CI-CDs.
- Depending upon my time commitment after the final evaluation of GSOC in August I would like to extend the project to set up CI-CD, good practices etc. as well for the project.

About me

I am a pre-final year undergraduate studying computer science at KIET Group of Institutions, Ghaziabad, India. My fascination with computer graphics and creative coding goes back to when I was in high school. I loved the power of computers to generate beautiful unimaginable visuals and sequences in simple patterns, mathematical equations and algorithms. You can visit some of the sound visualisers that I tried in my free time. Sound visualizer 1 and sound visualizer 2.

Apart from this I also published a hyper casual mobile game in my freshman year on the play store, named Missiles Go!

I also like participating in 24-48 hour hackathons and interacting with people and communities. I also have a few wins by my side in such competitions by working on real-world solutions using AR.

I discovered Processing after I stumbled upon the Coding Train youtube channel. Soon, I came to know about the world of OpenSource. I started my open source contribution journey with p5py, a Python port of processing and from that point I have contributed to setting up visual tests for the libraries, implementing features etc.

I have decent experience with Python, and web development along with the ability to quickly get started with huge codebases and new technologies, something that the OSS taught me.

I was able to put some meaningful contributions to p5py. You can check them below.

Intermediate Issues

- Refactoring the APIs to support multiple renderers.
- Formatted the whole database with autopep8 convention, taking care of errors etc.
- Setting up a CI pipeline using GitHub actions to run linters, and tests.

Features and Bug Fixes:

- Frame_count bug, where draw, noLoop and Loop were not implemented properly resulting in wrong frame_count behaviour.
- Added local storage module
- Processing API support in p5py.
- Added support for new APIs to render Pshape in 3d renderer as well.

- Greeting message fix on PRs.
- Refactor end_shape and begin_shape to support multiple renderers.
- Implement hex property.

Following are my contributions on the docs site of an LFX Mentorship organisation named open-horizon.

One of them is finding orphan pages on the docs sites: PR #159

Other PRs: Link

I have also contributed to other graphics related organisations, which can be found over on my Github and portfolio.

I am looking forward to working with you all during this summer and contributing to the py5generator repo as well shortly.

Thanks :)