> Introduction to Programming II - Midterm Report
> Student id: cl174

Outline the project

I'm choosing to work on the drawing app template, the main inspiration comes from one youtube video[1] I saw a couple of months ago where the author tried to use a Python gaming library called pygame[2] to randomly generate abstract art with a variety of styles. Those randomly generated pictures can be directly exported as wallpaper, the whole package looks so elegant as I feel compelled to make something similar using p5.js.

Aside from the abstract art extension, I will be also following the video instructions to work on the stamp tool and the editable shape extensions. Professor Edward already provided the initial sketch of those two extensions but I need to extend the functionalities from the template.

Regarding the coding techniques I will be mostly working on javascript p5.js library, perhaps I will use a little bit ES6 syntax as it makes things a bit easier when it comes to object inheritance and readability of the code. But again I will stick to the javascript eco system, at most using a few external js libraries.

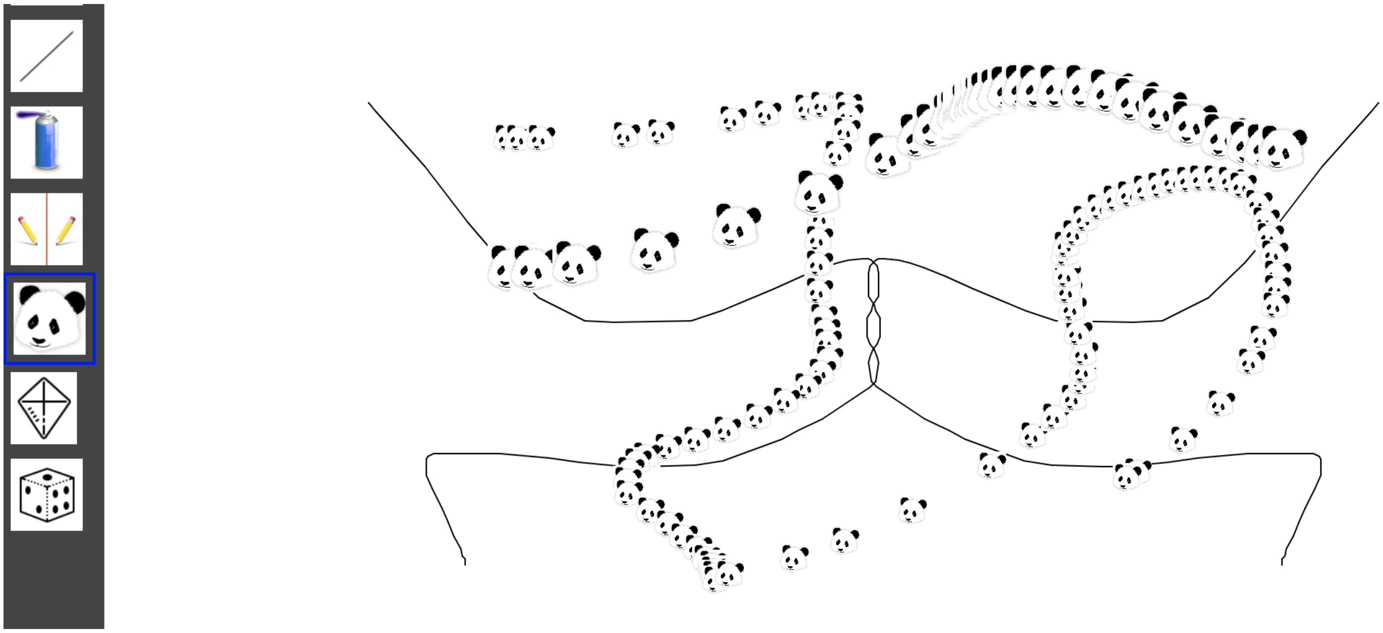
In my case the most difficult extension is probably the abstract art ones. The final python product[3] is visually appealing to me, but since I don't have much pygame programming experience, I probably need to read its API to understand how thing works under the hood before I jump into p5.js to build something on my own. So I am thinking the dvelopment should come to two phases. At first stage I need to read the pygame documentation to have an overview of its usecases and functionalities, then probably I need to read the source code of that python made abstract art generator as well. At second stage I will be building the abstract art generator using p5.js.

I am considering by combing the randomly generated art with basic drawing functionalities user could actually extend the random art to make something interesting using their creativity. Instead of drawing everything from scratch user could just click a button to generate something sophisticated enough then work from there.

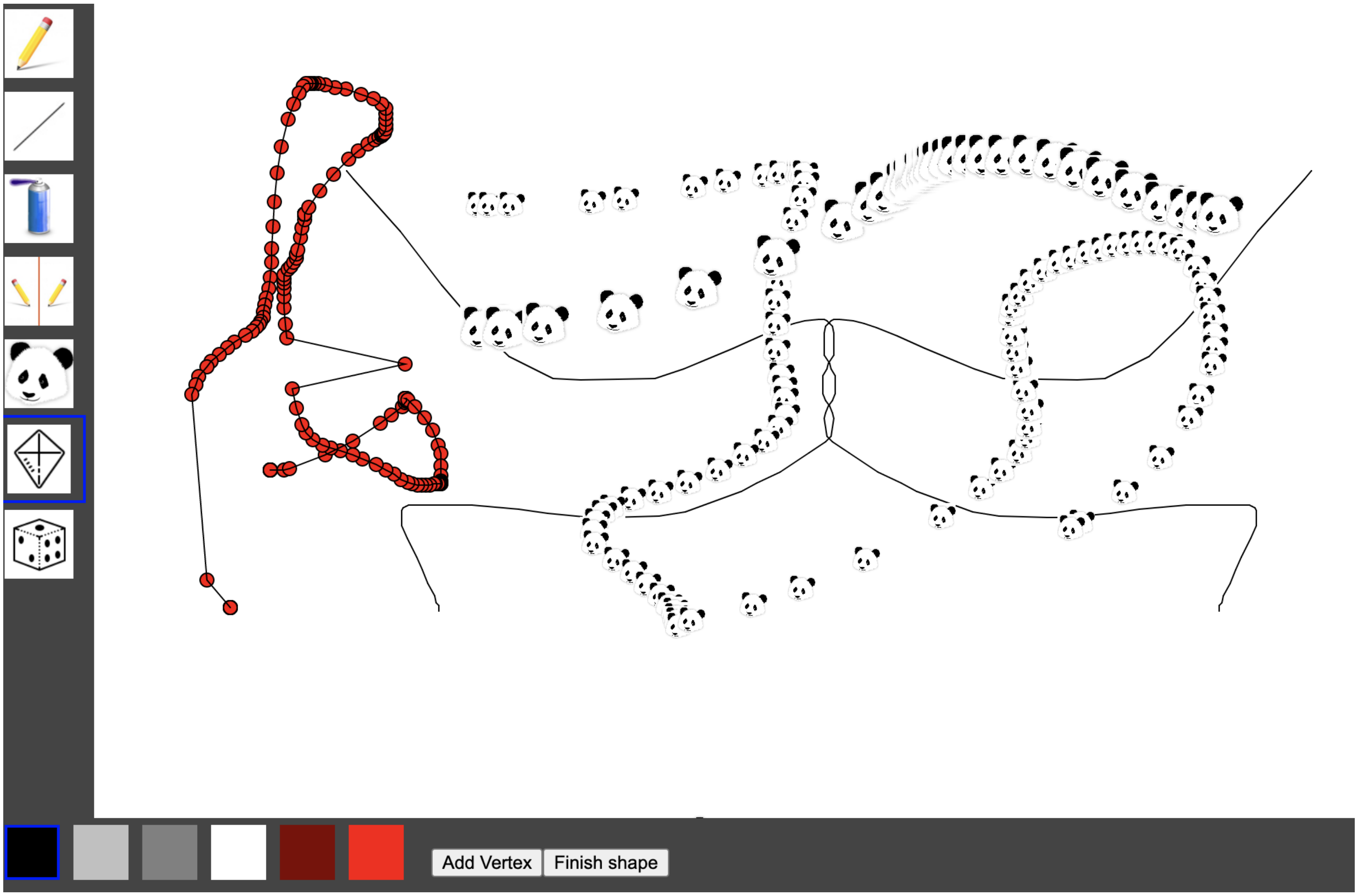
Progress so far

I have ported the stamp tool and the editable shape extensions to the template. I tweaked the initial code from here and there to make them work in the template.

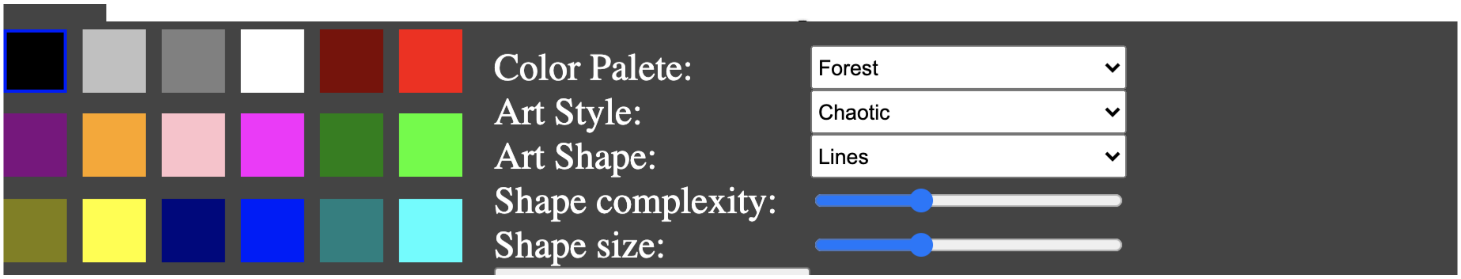
Instead of making a star stamp tool as illustrated in the video instruction I made a panda stamp tool where user could draw the head of the panda onto the canvas. User has the option to control the size of the panda to be drawn in the options panel.



For the editable shape extension, it's roughly the same compared with the app in the video instruction, user could add vertex to the canvas, also has option to drag the edge of the vertex to various shapes in editing mode.



For the abstract art generator, I haven't progressed too far yet, since most of my time were spent on the pygame documentation as well as reading the source code of the initial python built app. However I did get the user interface part working by using the p5.js api. I found it quite convenient to create something like a select box or a slider using P5.js, though visually it is not perfect. I probably need to write a bit css to make the user interface a little bit more appealing from the user's perspective. Regarding the user interface as I need to extend the app from the template, I probably have to put all the user selectable options into "options" div. I found in the template the web canvas were separated into different parts and the positions were set using css grid. Since I have many options for user to select in the abstract art generator, I think I would need to use either grid or flexbox to lay out those options properly.



Color Palete: Forest

Art Style: Chaotic

Art Shape: Lines

Shape complexity:

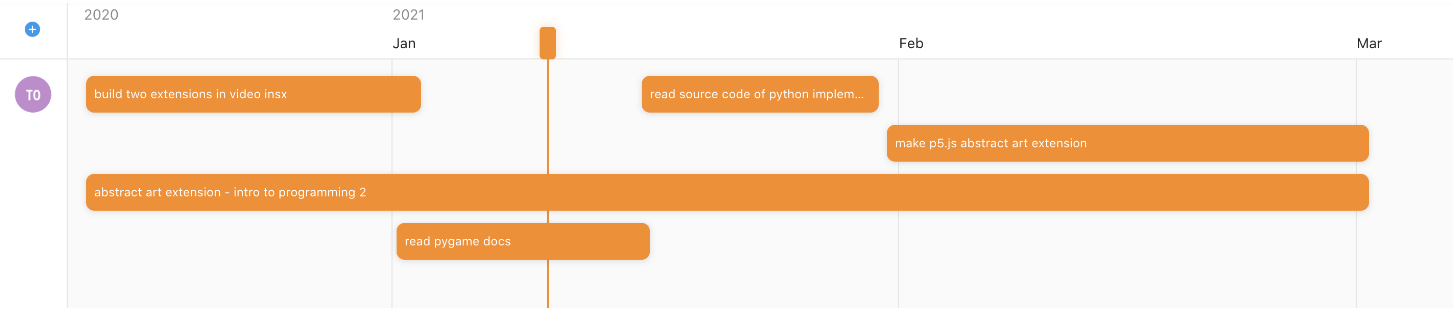
Shape size:

Time Allocation

As I outlined at the beginning I considered to divide this project into two stages. In first half I will read the pygame documentation, also read the source code of that python implementation to understand why thing works in the way it presented to be. At second stage I will be working on the p5.js implementation. It might be a bit risky considering there might be certain things pygame could cover but p5.js cannot. In essence pygame is a gaming library so it might has certain niche functionalities which p5 does not have.

Before the midterm at least I need to finish those two extensions that provided in the video instructions, those are the things I am sure I could do with a lille bit efforts. After midterm I need to work on the abstract art part.

I want to have something equivalent to the pygame implementation where user could easily generate some sophisticated random art shapes then be able to add their doodling to export as awesome wallpapers.



[1] "Showcasing Your Programming Projects - Code Jam Winners and Top Projects", Tech with Tim, Sep 24 2020, https://www.youtube.com/watch?v=NI5KqXasBx4&t=852s ↩

[2] "pygame", https://www.pygame.org/news ↩

[3] "Abstract Art Generator", Burakoli, 2020, https://github.com/Burakcoli/Abstract-Art-Generator ↩