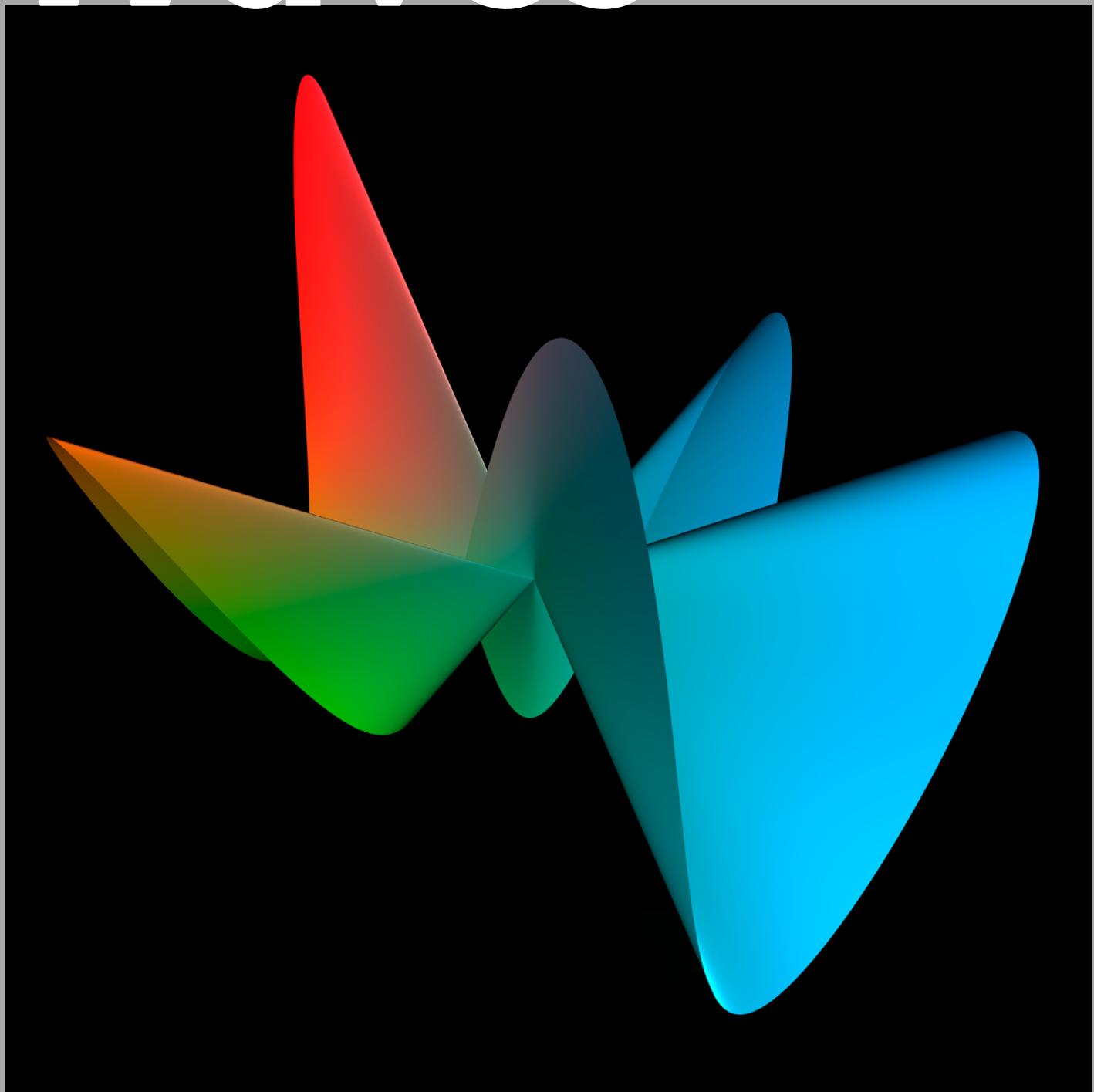
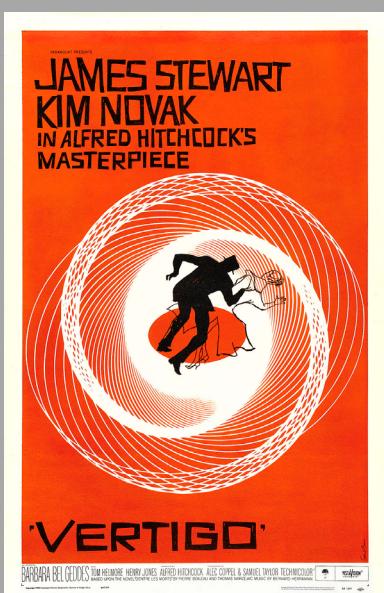


Waves



There are two types of waves. Mechanical and electromagnetic waves. Mechanical waves we can see at the ocean's shore or as ripples in a fountain. We can also hear them as sound. The second type (the electromagnetic waves) consist of oscillations of electrical and magnetic particles. They vary in wavelength and include radio-waves, microwaves, infrared waves, lightwaves and ultraviolet waves. X-rays and gamma rays are also part of that same group. Waves can also be generated with a decorative purpose in mind. For instance 'Vertigo' is an American psychological thriller directed and produced by Alfred Hitchcock in 1958. The title sequences and movie poster were designed by the American graphic designer Saul Bass. Both title sequences and poster are using Lissajous figures. Named after the French physicist Jules Antoine Lissajous (1822 – 1880). This chapter is about generating and modifying Lissajous figures.



'There is nothing glamorous in what I do. I'm a working man. Perhaps I'm luckier than most in that I receive considerable satisfaction from doing useful work which I, and sometimes others, think is good.'

Saul Bass, 1920 – 1996, American graphic designer and filmmaker.



MyCodeHistory: 8 February 2015

When I had a look at the result of this program the only thing I could think of was to change it's graphic appearance. So I began with renaming some variables to get a better understanding about what the program was doing. I thought it would be a good idea to strip down the total animation and only keep the most essential things in it. I think the graphic needs a title. For instance 'Harmonic oscillations with frequency x'. Where x stands for the amount of frequencies. And it would be nice when the frequency is updated (by hitting the 1 or 2 keys) that the x in the title updates too. Beside frequency we also have to deal with phase shift. So maybe its better to include that too and than make it a separate text combined with the frequency (measured in Hertz (Hz)). Phase shifts are typically measured in degrees where a complete cycle is 360 degrees. I colored that line red. Another problem which had to be solved is the positioning of the frequency text amount and Hz. When the frequency gets higher than 10 or 100 the number is going to run through the Hz characters. I used the utility function nf to display always 3 numbers for Hz. For phase shift I used the same nf function. As a last step I just put some more explanation to the graphic. And I found another useful line which connects the endpoints of the moving lines in the circle and timeline with each other. I will make that yellow.

Lets see if it works when I display the real degrees next to the moving dots. It seems that the AngleDegrees variable would do that job. I found that the rather strange number of 57.4 is the amount to be multiplied with AngleDegrees. I have no idea why this works. Now that both numbers are on the moving horizontal line I have one problem left. How to position the degree sign. When the degrees are smaller than 100 (containing only two digits) the degree sign is way too far from the number itself. I could not find any function which calculates the width of 2 and or sometimes 3 characters. So I solved that problem with the new

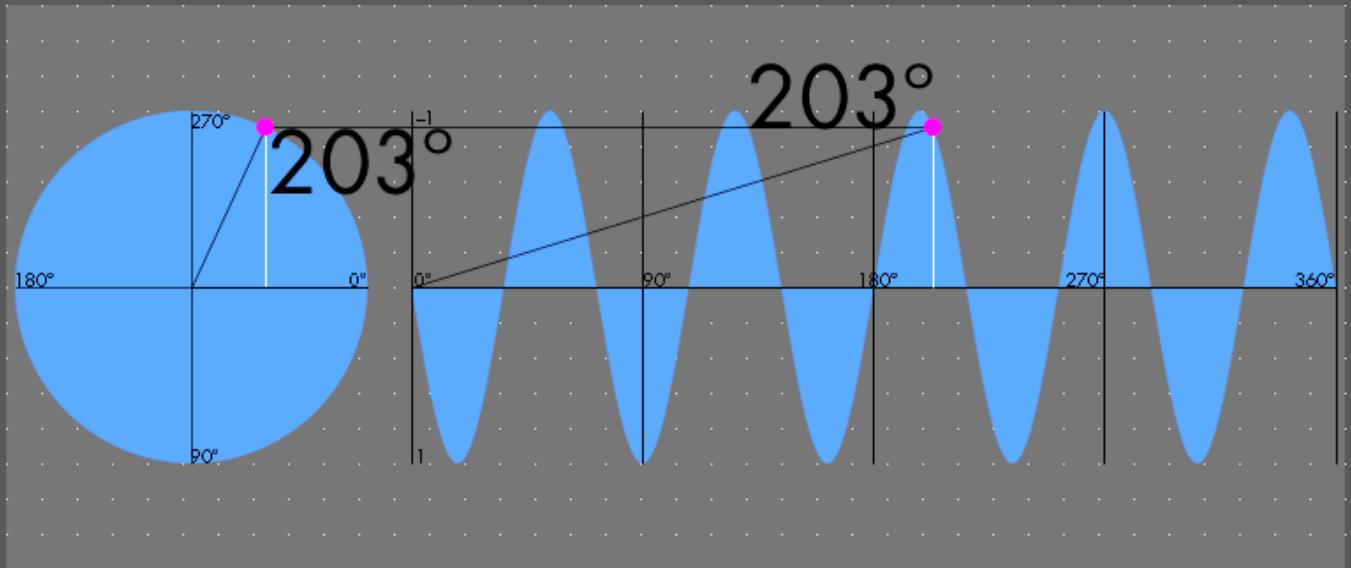
variable numberOfCharacters. And than it is simple a matter of writing an if statement. If numberOfCharacters is smaller than 100 than position the degrees sign on that position. Otherwise position the degree sign at this position. Finally I did not change much in these examples. It also did not lead to many variations. I replaced some colors and positioned some text. But it helped me to get a better understanding of what harmonic oscillations do and how they work.

Because this program is a scientific graphical demonstration of how harmonic oscillation works I have no reason to modify the graphics behavior. Harmonic oscillations were investigated by Nathaniel Bowditch in 1815, and more detailed in 1857 by Jules Antoine Lissajous. Harmonic oscillation is a scientific proven fact. And sometimes its better not to worry too much about those facts.

Haydn enjoyed traveling because it took him away from his difficult wife. One day a caller remarked on the large pile of unopened letters on Haydn's desk. 'They're from my wife,' the composer explained. 'We write to each other monthly, but I do not open her letters and I am certain she doesn't open mine.'

Franz Joseph Haydn, 1732–1809, Austrian composer.

Harmonic oscillations



Frequency

5Hz

Phase shift

0°

MyCodeHistory: 12 February 2015

This program is another visualization of a scientific fact: 'Lissajous figures'. The previous program used one oscillation in the x-direction. This program calculates two oscillations in the x- and y-direction. I have changed the global variable names and the size of the display window. I also added some word-spaces to improve the readability of the program code. Made the background gray and I filled all curves because the strokes were giving a bad image quality. I used only primary colors to fill the curves. And I'm looking for more information to make the page more interesting. In a later stage I decided to remove all fonts and imported a .png for the header. I did that because of the very poor character spacing in Processing. In the smaller point-sizes you won't notice that but in large font-sizes it is terrible. But there are more problems. I would like to reflect the settings (in numbers) in the text 'Horizontally frequency', 'Vertical frequency' and 'Phase shift'. If I move the dot-positions in the graphic I would like to see that the numbers change in the subtitles.

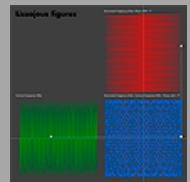
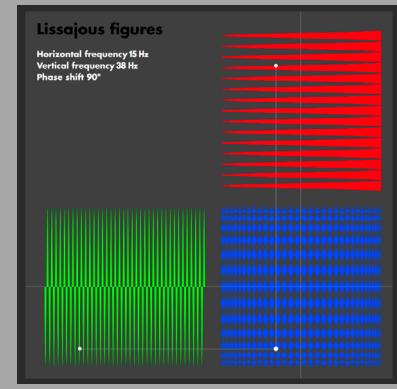
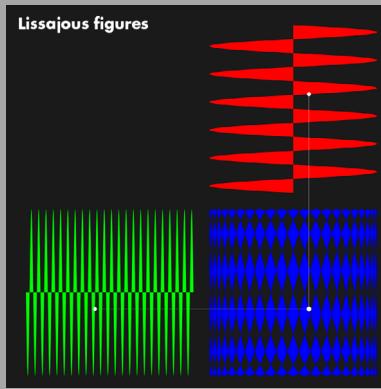
To avoid problems in JavaScript I added those subtitles in a background image. Which gives me no opportunity to change the position except to changing it in Photoshop. But than a major part of the font problem will be solved. I only have to position an off-font at the specific places in the subtitles. Tried that but with no success. It gives the same results as with the Futura.otf. So I stick to that font. Beside of that the programs work only in Safari and Firefox. I was searching for a Futura look-alike which was working in JavaScript. I found a font called leaguespartan-bold.otf. But that is not an ideal solution because the leaguespartan-bold.otf looks too bold when compared with the Futura bold.

I found a new option. Getting rid of the subtitles. I am only going to use the title 'Lissajous figures' in the background. Furthermore I give all other information about the curves at the white dots which are orbiting the curves. Because the zero point has been translated I need to re-calculate the positioning. The zero position is now located at point 600, 600 but after a while I thought these positions are not relevant. I just display the frequencies and phase shift at the white dot's positions. This animation is fine as long as you don't crank up the frequency rate of one or both frequency curves to lets say 7Hz. If it is higher everything will be unreadable. So I think its better to make that information static. But that is also more boring. Until here I don't want to go any further with these examples. I could make more variations but the intention is to understand how Lissajous figures work. And that is now clear to me.

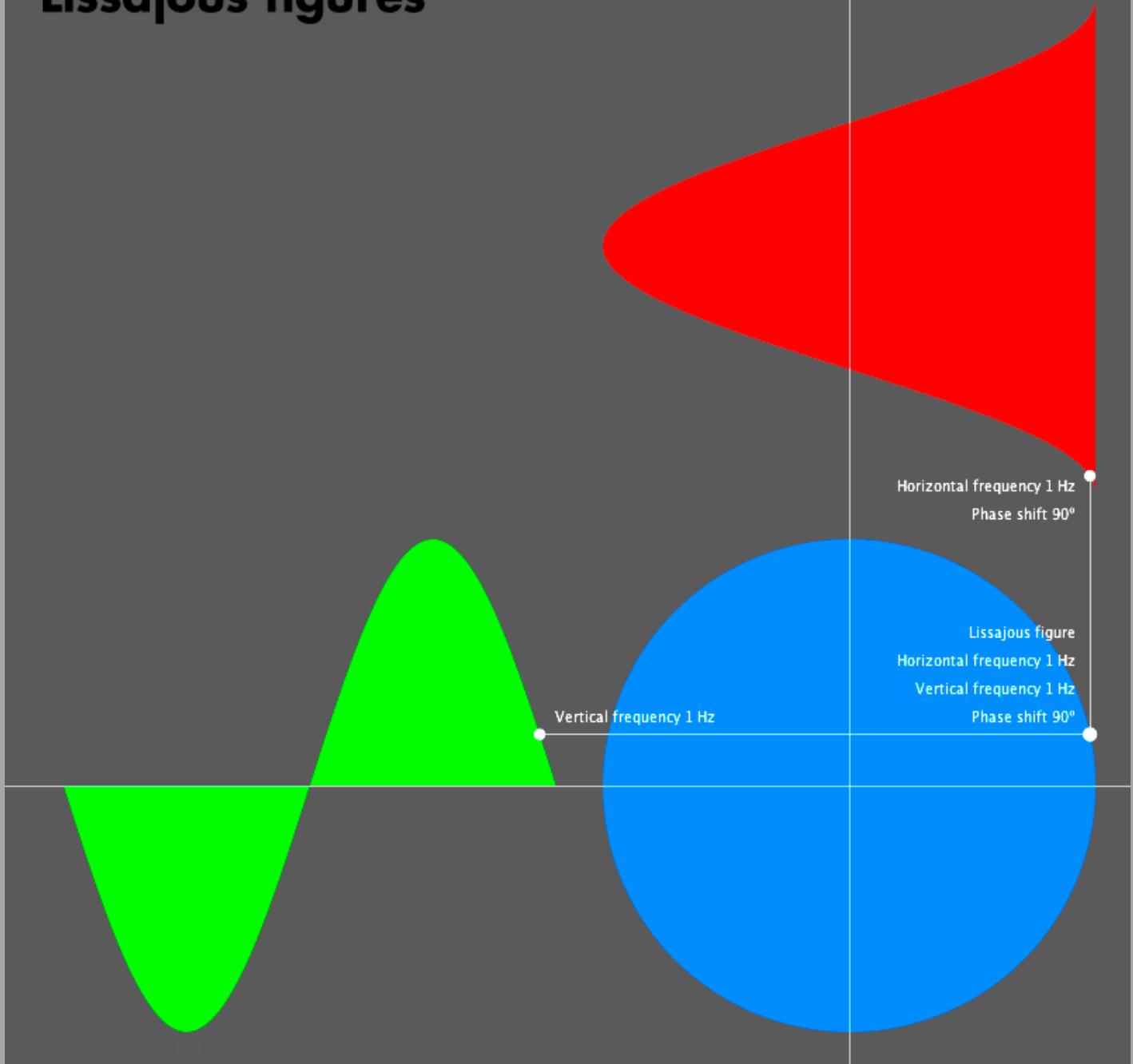
I am getting more worried about the Processing to JavaScript conversion. And when you start using fonts it is even worse. During this example I spend a lot of time on finding out how I could use fonts in JavaScript and Processing. These problems took me more time than getting to know how Lissajous figures are created. Beside of that there are several browsers which not fully support JavaScript. I hope these problems will be solved in the near future.

Do not spoil what you have by desiring what you have not; remember that what you now have was once among the things you only hoped for.

Epicurus, 341–270 BC, Greek philosopher.



Lissajous figures



MyCodeHistory: 14 February 2015

In the previous program I've worked with Lissajous figures. This program makes it possible to create even more variations by introducing modulation of the oscillations. I will not go into the concept of amplitude modulation because that is already perfectly explained in the Generative Design book. But in short two signals (waves or oscillations) are composed into one curve in which both signals are modulated. There are two versions of this program. Both versions use two signals. The information signal and a carrier signal. At first sight it was not really clear to me which line in the graphic represented the information signal, the carrier signal and the modulated signal. Looking at the code I could trace these signals back. So I colored the information signal red. The carrier signal green and the modulated signal blue. And I made all signals higher by dividing the height by two instead of four. Hitting the one, two, seven and eight key gives interesting patterns (as long as you keep hitting them). Maybe it's an idea to make the height very small. I don't know if it's against the laws of amplitude modulation but what if I make the signals (independent from each other) lower or higher? Would that make sense? It makes sense for the images. So what about attaching a few function keys to that idea. I used the up and down arrow keys for increasing and decreasing the line thickness.

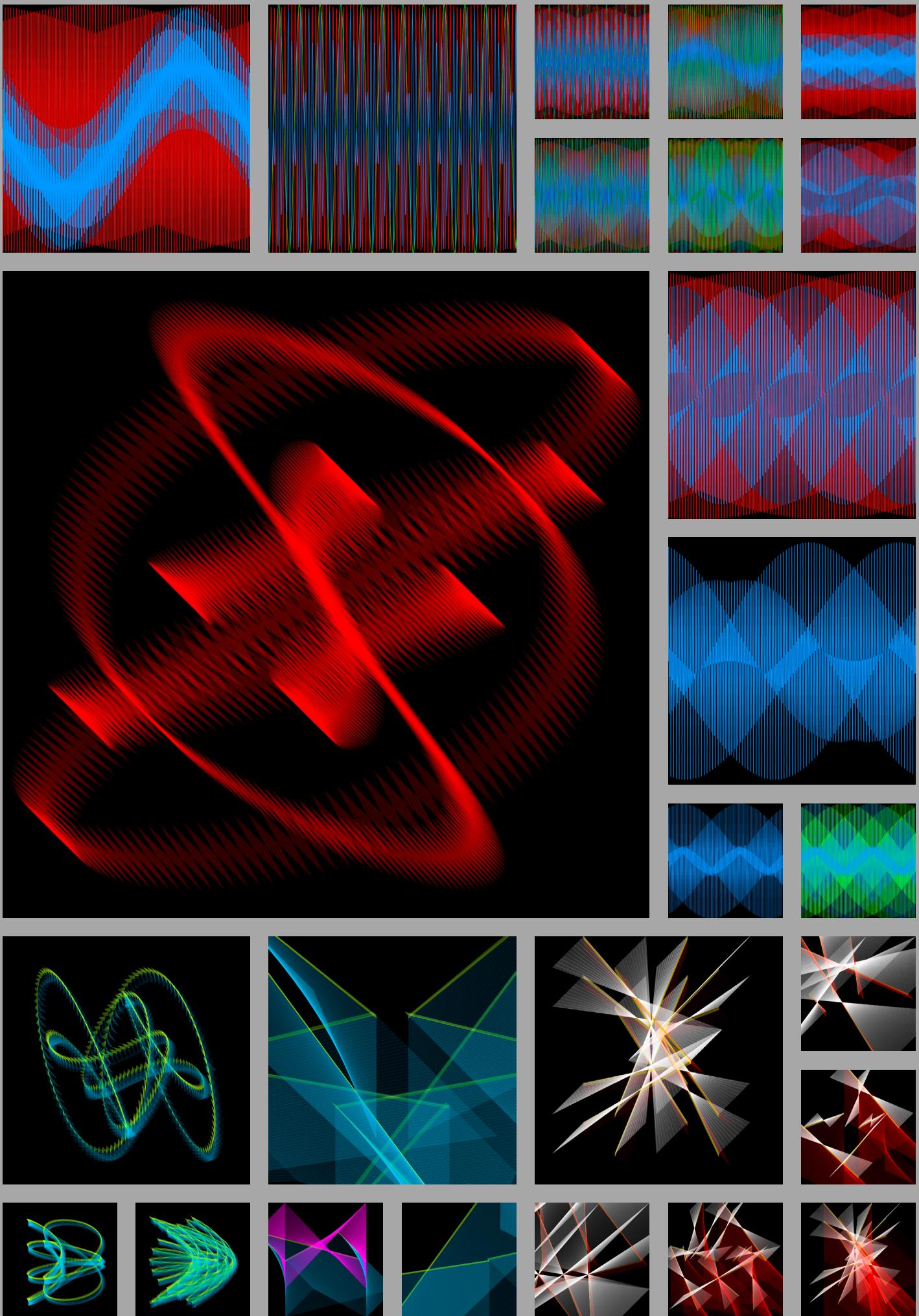
The second version of the program combines two modulated oscillations to define the x- and y-coordinates. Changed the name of modulationPhi variable into ModulationPhaseShift. But it seems that it is not being used in the program. So I commented it out. What does the local variable w do? Changed its name in WidthDistance. Not sure if that is correct. If it is not correct I will change it later. Anyway... I don't see the point of seeing points in a line. I commented out line 96. (PointCount = mouseX x 2 + 200). Hey! That is a line of code! Yes it is.

But I've commented it out. I've also commented out line 84, 134, 135 and 136. Introduced a new global variable CopyLine. Which copies lines in a for loop. I have always Matt Pearson's quote in my head from his book Generative Art (page 71): 'The next step—and I firmly believe that, if in doubt, this should always be your next step—is to multiply it all by 100.' Did that but 100 seems a bit too much in my for loop on line 137. So I stick with 40 for now. I also uncommented line 96. Made DrawMode one almost the same as DrawMode two. The color is blue and I left the CopyLine variable out only once in line 114. Added a range of yellow colored triangles to DrawMode one. Decreased PointCount from 800 to 10. In line 110 I multiplied Position_X x width. Although I am not working in 3d it gives a 3d impression. Decreased PointCount to only four points. But added a new for loop on line 12. This reduces the amount of possible figures to less than 10 per DrawMode. Increased it to 12.

I think that I was lost in amplitude modulation in the end. I was so far away from the original program that I was able to create totally different graphics. The waves are disappeared and they are being replaced by semitransparent planes which are accompanied by light brownish lines. You could interpret that as absent-mindedness but I rather call it creativity-light.

Bowles's usual daily ride took him along a road through a turnpike gate at which he had to pay twopence to the toll-keeper to allow his horse through. One day he passed that way on foot and tendered the twopence as usual. The gatekeeper, puzzled, asked: 'What's this for, sir?' 'For my horse, of course!' 'But, sir, you have no horse!' 'Oh, am I walking?' exclaimed Bowles.

William Lisle Bowles, 1762 – 1850, English priest, poet and critic.



MyCodeHistory: 19 February 2015

The next step is to program images into three dimensions. An additional sine wave is required to control the movement of points in the z-direction. The shapes that can be created are totally different if you compare them with the images of previous programs. And because you can look at the images in three dimensions you can see them at any angle you would like to. I imported the light settings from the previous program. And I also filled the gaps between the planes of the objects. I thought it would make better images. But after a while I checked if that conclusion was true. So I re-introduced the gaps in the objects again but a while later I thought it made the object less clear. I filled the gaps again.

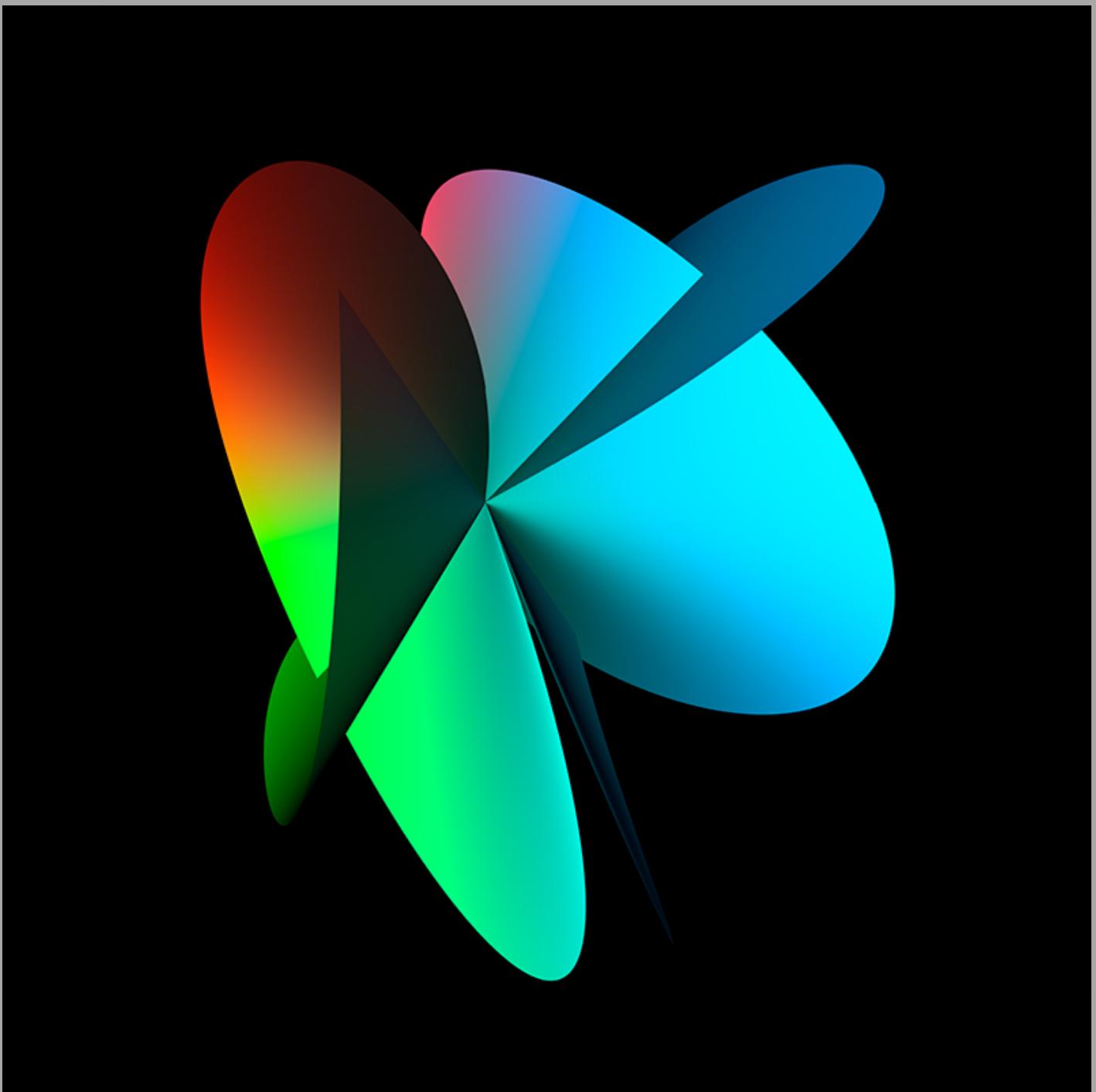
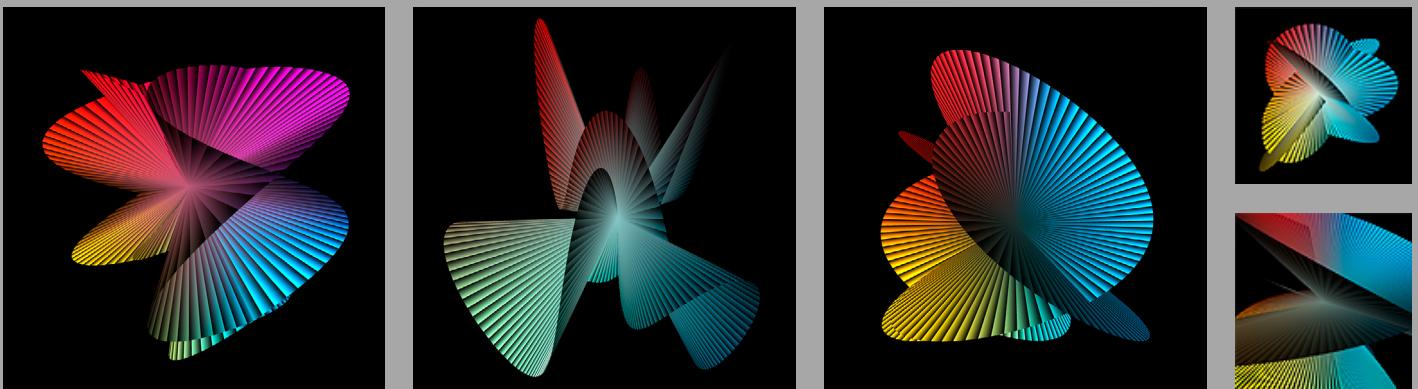
Started experimenting with looking to the shapes under different angles. And saw that TileSaver made a few mistakes. It saves a file in four parts. That's what it always does. But it positions the four parts in a random order. So it sometimes looks like I have three objects instead of one. I think that is fine (for now). But after some time I thought it is strange that TileSaver does chop up the images. When you run the program for the first time it doesn't do that. TileSaver says: '025.00% completed. 1/4 images saved. 050.00% completed. 2/4 images saved. 075.00% completed. 3/4 images saved. 100.00% completed. 4/4 images saved. Save: 1_4_2_0_0_1600x1600.png Done tiling.' Ok. Fine. Looks good. Now I hit the 2 key which increases the frequency on the x-axis. Run TileSaver again. Has this something to do with the function keys? Weird. In fact I could paste the image in the right order in Photoshop. But that is going to take time. I've put ImageSize on 3 but it keeps on tiling and delivering me a complete image in a wrong order. Ok. Than I have to live with that and repair it later. Fortunately TileSaver reacts very logical.

I think I need a zoom function to get closer to the object. But sometimes it is better to see it a bit further away. Used the plus and minus keys for that. I tried a few versions with outlines but that did not work. The images are not getting any better. And lines do not accept the color of the lights. Lines ignore lights. I have noticed that TileSaver is only working properly when you run it once and for the first time. And than you need to re-run the program. Otherwise it saves the tiles in a wrong order. Continued with the search for new objects and interesting combinations. I replaced the triangle_fan with triangles on line 141. That gives a surprisingly smooth and a totally different expression to the objects. Added a transparency of 50 which gave me a bunch of unattractive artifacts. So I lowered the transparency to 10. Which was not a good idea at all. I have decreased the PointCount from 2000 to 10. That gives very triangular objects. Which are (by the way) not very interesting. Used PointCount 20. Still pretty boring. I try 40. Even 80 is not interesting. What about 8000? Great!

Communicating with a computer via programming takes you often to areas where there is really nothing else to find than things you don't want to see or know. That may seem not functional. But I see that as a good thing. Because it only concludes that you are searching in the wrong place. It is similar to the fact that a lot of human to human communication is not functional. But human to human communication is even more complex than programming.

I absolutely hate people shouting to their smartphone on the train, 'Are you there? Oh, good! Fine! Great! I'm here! When are you there? I'm on the train, but later I'm there.'

Simon Michael Schama, 1945—, English Professor of Art History, Columbia University, New York.



MyCodeHistory: 26 February 2015

This program connects all points on the Lissajous curves with each-other. The further the points are positioned from each-other the more transparent the lines become. This program comes in two varieties: a program and a tool. In the program I've cleaned up the code. I don't pretend that I made the code better than it already was but it's now more readable to me. For instance: I don't know what the global variable ConnectionRamp does. I commented that out and nothing serious seemed to happen. I also found that the local variable h is not used. I commented that out too. And after a few more adjustments I decreased the LineWeight to 0.6. That seems silly. But it really makes a difference when compared to a LineWeight of 1.0. Just for fun I have left out the line connection. That resulted in a render-time of 14 seconds. Which is strange because the program does not have to connect the points. Knowing that I removed the points and re-introduced the lines again. But I left the for loop in, with the same settings as I used with the points. I introduced some color but the result was not what I had in mind but I think its worth to continue.

I started again with the initial program. I checked what I did wrong and why the image quality got worse during the previous session. I saw that LineAlpha is set to 100. That brings more detail into the image. I also removed the for loop which slows down the program. And that solved the problem. In the previous sketches I tried one example with points. Maybe I can combine those points with the image I already have. But no! Points did not work good because they were too prominent. So I replaced them by ellipses. I also would like to get away from those symmetrical shapes. So I replaced the TWO_PI by 90 radians. TWO_PI is a mathematical constant with the value 6.28318530717958647693. To summarize that: it is a full circle. 90 Radians is a quarter circle. Made some variations with the radians set to 180 (which is half a circle). Suddenly

a problem popped up with displaying images somewhere in the middle of the display window. But the problem has also something nice to it because it's not in the center. So I keep it like it is.

The second program is about using the Lissajous tool. During the renaming of the variable-names I came across an interesting variable naming problem. It is also strongly related to the typeface. Looking at the variable i1 (it could also be il) I was wondering what it did and what it represents. By doing a search within the program I thought that il was an abbreviation for illustration. But the search-dialogue did not find il as a variable. So it must be i1. In addition I found also an i2. It is used in a while and a for loop. I assumed it had a connection with creating pdf-files. So I created a pdf-file. I'm not sure but I think that i1 stands for the begin or starting points and i2 is the total amount of points (which is in fact the same number as the global variable PointCount). So I renamed i1 to StartingPoints and i2 to totalPoints (starting with a lowercase because i2 is a local variable). I think when you enter the calculateLissajousPoints function the variable names are getting very abstract. Float t, x, y, rx, and ry. What do they do? What does t do? It seems t is not used. So I comment it out. In the drawLine function there are also three variables of which I don't know what they are doing. But I found out that d is the distance between points, a is the transparency/alpha value and h is the hueValue. Changed their names to make the program more clear to me. Now all that is left is to change the names in the controlP5 interface according the global variable names. Otherwise the controlP5 interface will not react on any input.

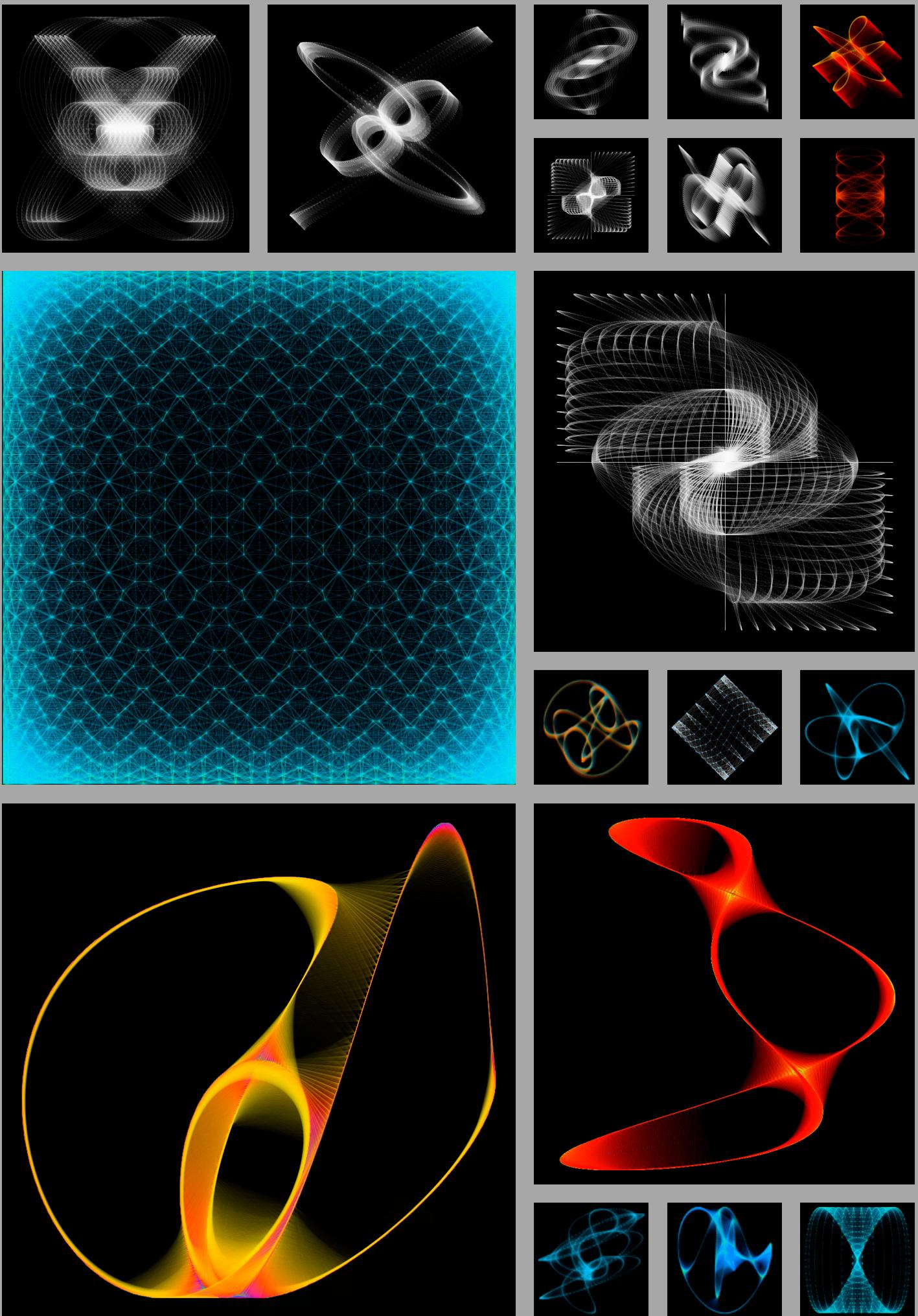
I started to create these images by using only the Frequency_X and Y and PhaseShift sliders but it seems that adding PhaseShift makes not much of a difference for the image. It just shifts it. So I added ModulationFrequency_X and ModulationFrequency_Y to it.

And that does make a difference. The best images are generated when Frequency_X and Frequency_Y are very low in their numbers. PhaseShift can be used to make the image a-symmetrical. ModulationFrequency_X and ModulationFrequency_Y are also best in their lows from what I have seen until now. I would like to check if it is possible to change the ranges of the controlP5 sliders to match them better with my preferences. If that is possible I can have more possibilities within my preferences range. Sounds complex but if I never use a higher point count as 1500 why should my slider go until 8000? It would be better to let the slider cover the range from 0 to 1500. Fortunately that is easy in controlP5. Just enter the number range in the lines which cover the sliders in Processing's GUI tab. Another thing is that I now can decrease the length of the sliders which gives me more space to judge my image.

ControlP5 is a very interesting library. It gives you a lot more possibilities on top of the things I ever might learn about programming. But it also makes the images more general. You might say: 'Hey! Anyone can pull a slider or check a checkbox!' Still these examples were very helpful to learn more about the controlP5 library. I will certainly use controlP5 again in the future.

Science is spectral analysis. Art is light synthesis.

Karl Kraus, 1874 – 1936, Austrian, satirist, essayist, aphorist, playwright and poet.



MyCodeHistory: 3 March 2015

This program is a drawing tool. The functionality of the tool is based on the fact that the previous program's method of connecting all points is not limited to Lissajous figures only. It can be applied to any set of points. There are two versions of the tool available. One for drawing with the mouse and one for drawing with the pen tablet. In this case I used the pen tablet tool. I always work with a pen and tablet because I think it works more accurate than drawing with a block of soap (mouse).

To begin with I changed some of the global variable names and a few names within the program so that I could get a better understanding about how the program worked. But later I realized that because this program is a tool it might be a good idea to just use it as a tool. And leave the program as it is. So instead of concentrating on changing the program I concentrated on working with the tool. But there is one interesting thing. When I reset the program I get an:

```
March 01, 2015 12:29:08 PM
controlP5.ControlBroadcaster
printMethodError SEVERE: An error
occurred while forwarding a Controller
event, please check your code at reset
java.lang.reflect.InvocationTargetException.
Exception. Why is this interesting?
Because the program keeps still running
and is fully functional. So I don't bother
and continue working.
```

But back to the drawing tool. What do I draw? I could trace images which you can import in the background. But I think I better try something else. How about just drawing lines? I started with drawing horizontal lines. Later I added vertical, diagonal, and crosswise lines. Not straight lines but manual. In later variations I did use a ruler to make straight lines. And when you can use a ruler you also can use round objects to trace with the pen tablet. Half circles. And some variations with a Linex A6036. I ended with drawing just random points by hand.

Thinking about what to draw before you actually start drawing always slows down the creative process. Critical thinking during the drawing makes you progress in ways you never had thought of before you started the drawing process. Or at least that's how it works for me.

All generalizations are dangerous, even this one.

Alexandre Dumas, fils, 1824 – 1895, French writer and dramatist.

