I took influence from a few different artists that are well versed in creative coding. One of them is called David Cool. I noticed that he produced some pieces which involved organic movement like circles which is what I intended to head towards. His work is somewhat different from mine because it involves circles that get bigger with each draw. However, with mine, the circles remain the same size but differ in place and thickness on the canvas. Dan Shiffman is another inspiration I used to get started with this project. I think he was the most helpful resource I found, simply because he explains every little bit he does, plus he uploads videos and tutorials onto his YouTube to go into further detail. If I was ever stuck, I would visit his website and YouTube channel as there was a high likelihood he would have done something similar.

The development of my project all happened rather fast as once I got the ball rolling on what I wanted to do, I went on a coding rampage. I started with our first project which was making shapes appear randomly on the canvas with different sizes and colours. This was the first thing that intrigued me because the idea of creating something that was so dependent on randomness was fascinating. I combined that with the idea of making code that draws shapes. I went on to look at different shapes and decided on a circle since it needed more code and would look more impressive (in my opinion). I also had to look into mapping sin and cos curves which I first had to understand. The first problem I ran into was making the code draw a perfect circle. I started off with using the x and y coordinates to set the distance and the limit it could travel, so it was confined to the canvas size. That was the first step, the second one was to remove those constraints on the circle and allow it to appear and draw anywhere on the canvas- even if it wanted to come off the canvas. To do this, I had to set the x and y coordinates to a reasonable random number, allowing it draw in-between those coordinates. After that, I wanted it to stop drawing the circle after it had completed it. This was by far the hardest part. I first trying out a percentage loop, where when the sin value reached close to 100% completed, it would stop drawing the circle and move to another one. But the closest it gets is 0.999 (1 is completed), it would jump around the canvas and draw a small ellipse and then move onto another circle. To stop this, I increased the number from 0.999 to 0.9999 to leave an even smaller room for error. One thing I would like to add to this piece is to make it a 3D space, so that cylinders are drawn instead of flat circles.

<https://github.com/Josh236/DAT405.git>

**References:**

<http://davidcool.com/portfolio/digital-painting/>

<http://shiffman.net/learning/>