Iteration[1]: Written Response

By Runxin Zheng

Remaking the project Squishy Stack by Ivan Rudnicki has led me to reconsider how shapes are formed and the elements that construct shapes, such as nodes, vertices, and curves. For instance, an ellipse can be seen as numerous dots lining up in a circular direction. From there, I started exploring approaches to creating an oval shape, from using the ellipse object in p5.js to adopting different math equations to drawing with curves. Each approach offers certain possibilities while posing certain limitations. To recreate the disruptive effects in the original work by Rudnicki, I experimented with different methods to modify these ellipses, and one of the learnings I had was that shapes formulated by equations are more rigid than shapes drawn from connecting nodes and curves.

One of the technical difficulties in using p5.js is understanding JavaScript and its terminologies. Many functions in JavaScript originate from mathematics so I had to understand the rationale and the relevant mathematics behind each feature to use it properly or creatively.

What was unexpected about my iteration 1 was that this project involved not only JavaScript but also physics, drawing inspiration from the natural world. My coding iterations started with the movement of a spring, which involved many factors in Newton's second law, including mass, forces, velocity, and acceleration. In the process, the visualisation of Newton's second law enabled me to better understand physics as well as organic, natural motion graphic design.

In the next stage, I'd like to explore different physics equations from the NASA Glenn Research Centre in the p5.js environment. The iterations will be conducted by drawing ellipses and applying different physics formulas to these ellipses. Through the experiments, I'd like to answer the following questions:

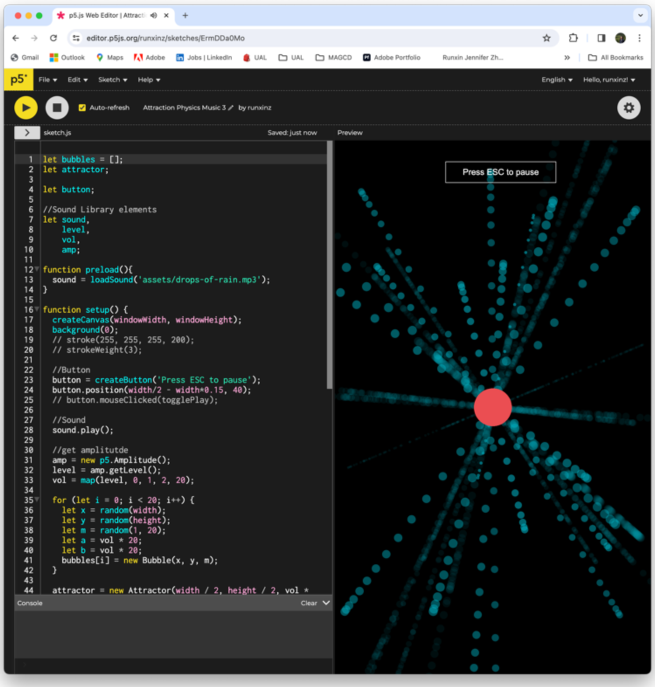
- What types of visual effects and reactions can these physical laws generate when written in p5.js?
- Do these physical laws always generate organic, natural movements?
- Can they also cause distortions or disruptions to shapes?
- Do these equations have different visualisations when applying different drawing methods?
- What are the limitations of using physics equations in a coding language that is designed for visual art?

Iteration[2]: Written Response

By Runxin Zheng

Latour (1986, pages 1-40) urges that one should consider two aspects when considering the origin of modem scientific culture, one of which is the information, the knowledge, and the content, and the other is the tools or mediums that bear the information and widens the reach of this information.

We can interpret the interface of p5.js Editor with these two aspects. The coding panel stores all the information – the values, the physics equations, and all other data. Coding can be understood as a technical innovation from inscription; however, it still functions like a human brain, where the information is encrypted and not yet mobilised. The preview panel is the medium where the mobilisation of codes takes place. When translated into a visual language, the information becomes colours, shapes, and motions that one can experience without necessarily understanding any coding languages.




To increase the power of coding, the language must be more accessible to a broader audience. An established coding language often has references, libraries, examples, and community-initiated sharing platforms. As a new user of p5.js and coding in general, I had gone through these references before I could communicate in JavaScript and applied them to my iterations to produce new images. My actions can be understood as a displacement of original references, just like the inscription, described by Latour (1986, p.16). Through modifying and creating the JavaScript codes, I've mobilised them onto a new dimension.

> Without the displacement, the inscription is worthless; without the inscription, the displacement is wasted. This is why mobilization is not restricted to paper, but paper always appears at the end when the scale of this mobilization is to be increased. (Latour, 1986, p.16)

Physics plays a crucial part in coding languages. In p5.js, the physics-based equations enable the simulations of real-world phenomena, applying the laws of nature to the virtual world. Coding, on the other hand, has brought physics into a new environment, no matter whether it is a screen, a projection, or an interface. In the world of p5.js, force becomes an ellipse that moves perpendicular to the bottom of a window; velocity becomes the falling duration of the ellipse. The traditional inscription of physics has been transformed and displaced, again.

In iteration[2], I've learnt how to simulate physical phenomena, such as free falling, friction, gravitational attraction, and spring. Since I am new to JavaScript and haven't worked with any physics formulas since high school, iterating in p5.js has been a slow journey. I was forced to think, observe, and eventually understand how things move, accelerate, decelerate, and stop. Physics equations tend to be firstly simplified and then applied to JavaScript though the fundamental logic of physical laws, such as Newton's Law, is still at play to make the movement natural to the eyes. In a way, the application of physics in codes is also an act of translating.

Once managing to create these simulations in p5.js, I then added a royalty-free song and converted its amplitude into gravity. Now, we not only hear the music but also see its 'weight' when the volume is encoded to a numerical value, then applied as gravity in a mathematical equation that draws all the ellipses and then drives them away.

What I learnt during these iterations is that sounds, such as songs and sound effects, can be read as data in coding but also as physical vibrations in the natural world. Based on this realisation, I wanted to see how p5.js can demonstrate the relationship between sounds and physics in a virtual, alternative way. How would this coding process enable me to understand motion graphic design differently? How would this visualisation change viewers' experience of physics and music? Will these movements become unnatural or even disturbing to our eyes?

In iteration[3], I reversed the input, music, and output, the physical simulations, attempting to 'hack' the primary use of p5.js and create an instrument or a synthesizer using this language. How would natural phenomena manifest through sounds or notes? Are these simulations an accurate translation of the real world? Is there anything that is lost in translation? At the end of these experimentations, I do think something has changed, perhaps lost, during these translations, which are likely to be rhetoric elements mentioned by Spivak (1993). After all, real-world physics involves more than just mere numerical values and equations. However, since the logic is still present in these simulations, the translation still creates new, valid experiences that could reshape our perspective of the real world.

Reference

Latour, B. (1986) 'Visualisation and Cognition: Drawing Things Together', *Knowledge and Society Studies in the Sociology of Culture at Present, Vol. 6*, pp. 1–40. Available at: http://www.bruno-latour.fr/node/293 (23 Jan 2023).

Spivak, G. (1993) 'The Politics of Translation'. Available at: https://pierre-legrand.com/16spivak.pdf (1 Feb, 2023).