Voice Over Script

This project began with the brief *Methods of Iterating*, where I recreated Karel Martens' silkscreen circle monoprints. I focused on repeating, layering, and rotating shapes—working manually, but intuitively. That iterative process led me to experiment further, cutting up silkscreen prints and recombining the fragments to see what new forms would emerge.

These early tests felt playful but inconsistent. I began wondering: what happens when randomness becomes a tool in itself? Can a design system incorporate chance while still producing meaningful, structured results?

To explore this, I moved into p5.js—a coding environment for generative design. I started by building random shape generators. The code handled placement, rotation, colour, and shape type—while I observed the outcomes. The input became a set of parameters; the output, a shifting visual field of unpredictable combinations.

This helped me detach from the need to make aesthetic choices manually. Instead, I was designing conditions.

From there, I wanted to understand the tool more deeply, so I created a series of code sketches. Some relied on freeform randomness, while others introduced grid systems or conditional behaviour. This technical exploration was essential—it taught me how form, behaviour, and constraint interact.

While exploring p5.js, I discovered Karl Nawrot's *Breu* typeface. He used architectural stencils not to reproduce buildings, but as tools to construct letters. This inspired me to think: what if I treated the random shapes from p5.js as a visual toolkit—as my own stencil set?

This is the first p5.js coding sketch I used. I began generating a custom alphabet using these modular shapes. Every glyph was a composition of parts—assembled from fragments rather than drawn by hand. The code had created the tools, and now I was repurposing them.

For the second exploration, I tried to identify the letter parts within the coded visuals themselves. This also resulted in an interesting set of typefaces.

Then I looked closer at Nawrot's references and learned that architecture played a critical role in his design language. That pushed me to revisit architecture in my own cultural context.

So I turned to Dancheong—the ornamental patterns found on Korean traditional wooden buildings and artifacts, especially temples.

The eaves of traditional Korean architecture play a significant role in framing views of the external scenery from inside the house. Dancheong reflects the dominant colours in nature: green from leaves, brown from trees, and shades of red from flowers. These complementary hues blur boundaries, creating the illusion that the landscape continues into the interior.

As a child, I often visited Korean temples with my family. We are Buddhist, and those vivid patterns—layered with colour and repetition—feel like home to me.

I extracted visual forms from Dancheong motifs, reduced them to modular parts, and began designing a new typeface. These shapes were no longer randomly generated—they were culturally grounded, formally structured, and emotionally meaningful.

This was a turning point, where I moved away from p5.js and focused fully on the architecture-inspired type system.

Generative coding taught me to see design as a process of defining constraints and discovering outcomes. Dancheong extended that logic—offering a rich, historical visual system to work within. What changed wasn't the method, but the material. I wasn't stepping away from generative thinking, but embedding it into cultural form.

My line of enquiry evolved with the work. I wasn’t just exploring randomness anymore—I was asking: how can structure and intuition coexist? What does it mean to create design systems that are personal yet procedural?

Articulation of Forms became a project about making through translation—between code and culture, between systems and memory. And in that process, the act of designing became a way of thinking.