

Process:

We began by searching for Arduino projects in general and brainstorming ideas using a mind map. During this process, we developed the idea of a pong game and explored the possibility of combining Arduino with SuperCollider, using the latter to generate sounds and control lights for the buzzer and LEDs. After identifying the materials needed for the game box and Arduino kit, we went to shops to collect everything required.

Once we procured all the materials, we divided ourselves into groups of two. One duo focused on the coding aspect, another group went to the laser lab to measure the MDF board for the pong game and paint it later on, while the last group worked on assembling the Arduino kit and setting up the LED lights. We used a trial-and-error method to refine the code.

Arduino Pong Game Code:

The Arduino code in the first file produces a basic 1D pong game in which two players compete by controlling a virtual ball using buttons, with scores displayed on an LCD. The game's logic is simple, with functions handling key aspects such as ball movement, scoring, and resetting the game. LEDs show the ball's position, and the ball moves between players in response to button presses. As the ball's speed increases over time, the difficulty of the game also rises.

We found it fascinating how hardware elements like buttons, LEDs, and an LCD display were used to provide players with instant visual feedback. Working with physical components in Arduino projects can be satisfying once everything is set up correctly, though it can sometimes be challenging due to hardware limitations or wiring issues.

SuperCollider L-system Sound Code:

The second file delves into a more imaginative and abstract area: sound production using SuperCollider's L-systems mathematical approach. The two rules in the code form a basic L-system that controls how strings evolve over a number of iterations. These iterations are then mapped to sound parameters, where each character ('A' or 'B') denotes a particular frequency and duration of a sound wave.

Although this project is more focused on generative art and sound creation, it feels very distinct from the Arduino hardware project. Real-time audio synthesis in SuperCollider is both challenging and exciting because it allows us to directly manipulate sound production using mathematical structures. We found it interesting how something as abstract as an L-system can be transformed into a concrete output like music. The clever mapping of characters to sound attributes such as frequency and duration gave us a deeper understanding of procedural sound creation.

Linking the Two Projects:

While the Arduino Pong game and the SuperCollider L-system project explore interaction in different ways, they both rely on rules to define interactions.

- In the Arduino game, the rules govern the ball's movement and scoring based on button presses.
- In SuperCollider, the rules define the structure of the L-system, which in turn determines the sound patterns.

Both projects create an interactive loop—one in which players engage in real-time through physical buttons, and the other in which sound evolves through code. This demonstrates how programming can facilitate interaction in both auditory and physical environments. We are excited by the potential to bridge these different types of experiences. In future projects, we might consider applying the structured, rule-based approach of the SuperCollider L-system to more complex gaming scenarios or explore how game interactions could influence dynamic sound generation.

In conclusion, these two projects demonstrate the diverse possibilities of creative coding—from interactive games involving real hardware to immersive soundscapes generated through procedural algorithms. Whether controlling tangible components in Arduino or experimenting with sound synthesis in SuperCollider, each environment presents unique challenges and valuable learning experiences.

Documentation:





