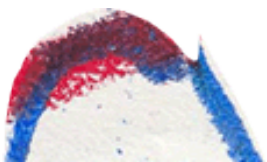
Configured Performance Instrument (CPI)

a performance insturment that relies on chance and improvisation.

By Aous Hamoud

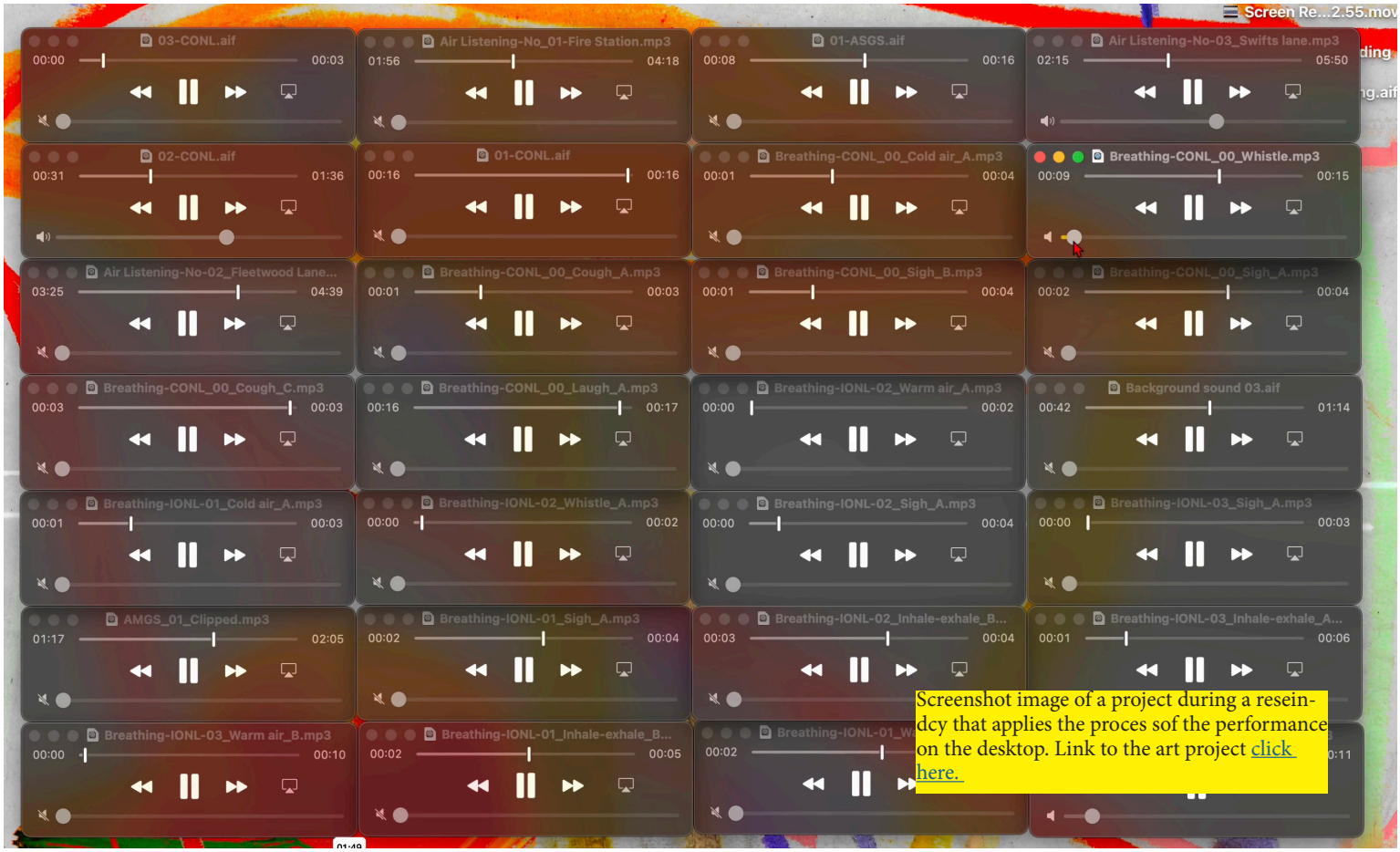
Link to video demo

https://www.youtube.com/watch?v=wGhuCk5RVlo



Introduction_

Configured Performance Instrument (CPI) is a project that was built upon a set of rules taken from a performance process. Its conceptual task is to find a way to connect between sound and drawing, as well as its relation to chance and improvisation. The instrument consists of nameless independent shapes on screen that have their own memories (sound) floating around and can be encountered by the user.



Screenshot image of a project during a resein-dcy that applies the proces sof the performance on the desktop. Link to the art project click here.

Concept & Background Research_

The reason for making this project came from past performances I have participated in, both with different artists and individually. In these performances I tend to share my recordings, looking on the use of sound as a research tool, I lay out the tracks on the desktop using QuickTime and play everything at the same time. This listening method allows different layers to appear through coincidence. Then I start changing the volume and embrace chaos through improvisation.

I took this process and developed it into a tool for improving and expanding my performances. At the same time, drawing has been an ongoing practice for me as a way to reflect on memory. This project became inspiring because it allowed me to connect both practices, bringing sound and drawing together as a single performative tool.

I titled the project Configured Performance Instrument to reflect what I am attempting to build: an instrument rather than a fixed artwork. Using p5.js, I explored a way to integrate my drawings and sounds simultaneously, allowing visual and sonic elements to operate together as a live system.
Configured Performance Instrument recreates my performance logic in a form that is accessible to other users, helping them understand improvisation and the importance of relying on chance.

At its core, the project is time-based, as it incorporates previously recorded situations collected from different times and locations. By reintroducing these recorded sounds into the present moment, the performance creates a window into memory, allowing different memories to intertwine with one another. Each sound carries its own past, and when activated in the present, these fragments meet by chance.

Technical Implementaiton_
Phase One - On Paper

- ## At the beginning

This is the beginning, where I was planning how to implement the sounds and collaged objects. First, I developed the concept that each sound would have its own shape, and that all of them would float and intertwine with one another. I began asking myself questions in order to build an algorithmic logic and create distinct personalities for each object.

Here are the questions:
What is going to be created over and over again?
How is it going to be encountered?
Where does it begin?
Does each object play one sound, or does it choose randomly?
Do objects collide, or do they have their own layers? If they collide, what happens and which object is affected?
Do objects bounce back from the edge of the canvas, or move in a wrapped motion?
What happens when the user inte[redacted] objects?
Is there a beginning and an end?
When does 9t crash or stop?
Does it crate more shapes?
Amplitude close and far!

- ## Explain the project

Some of these questions helped shape an understanding of what I wanted to do. I took the algorithmic thinking of the performance and integrated it as a plan for creating the programme.
This diagram explains how I transformed the activity used in the performance other create a plan for the P5.js code I will make.

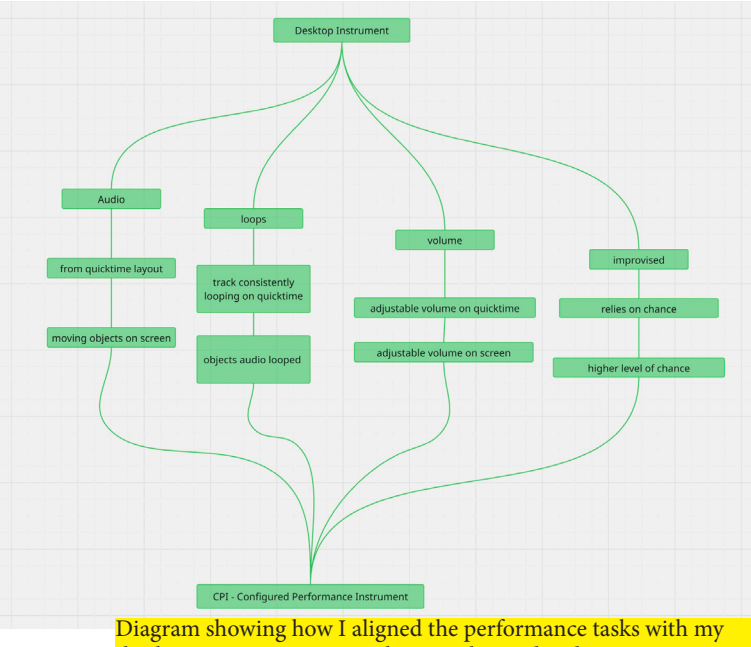
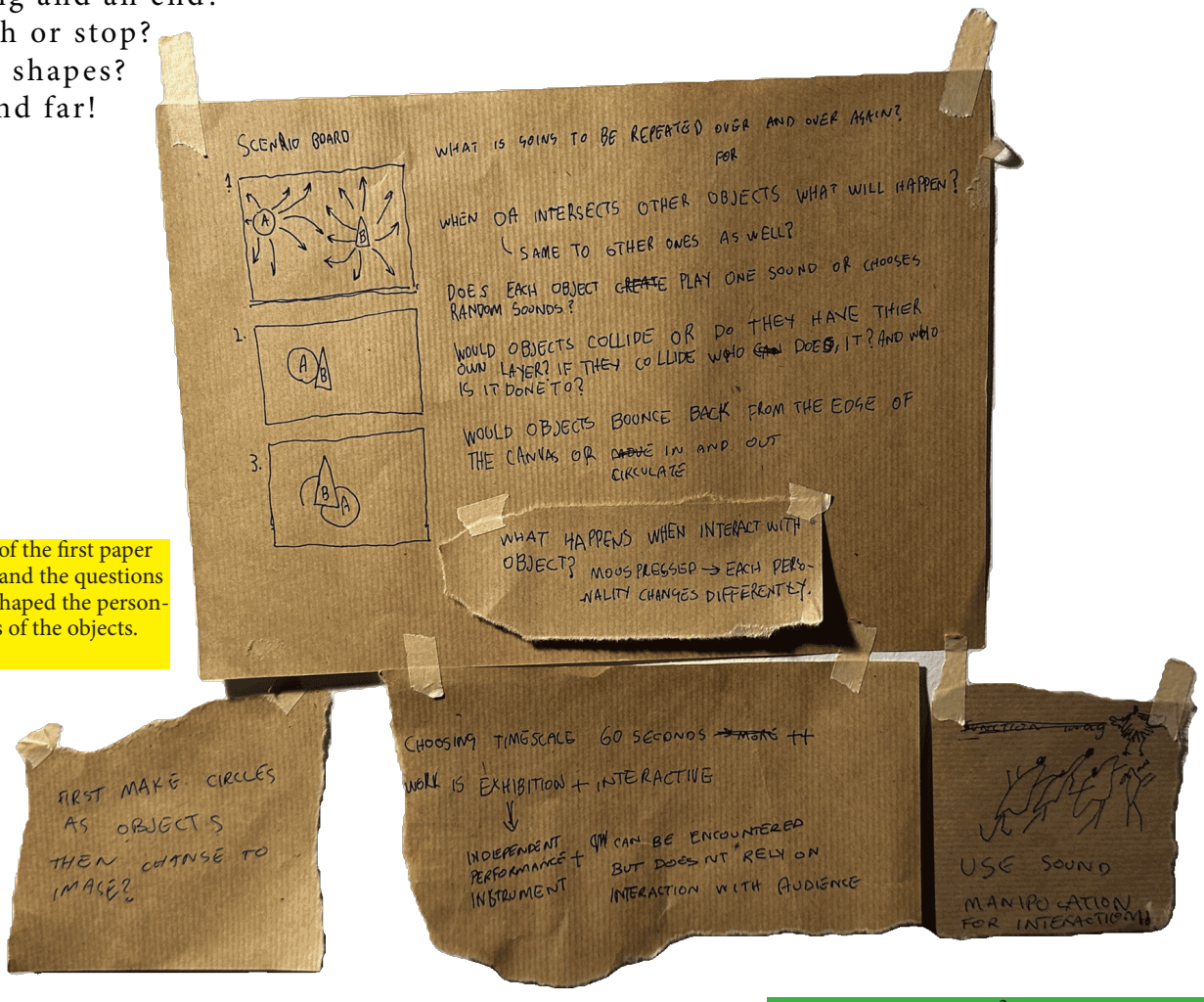


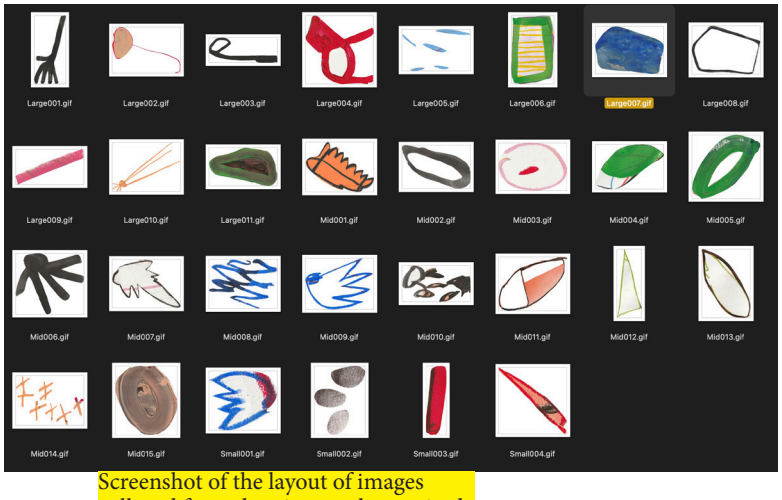
Diagram showing how I aligned the performance tasks with my [redacted]ocess to initiate the CPI plan and code.



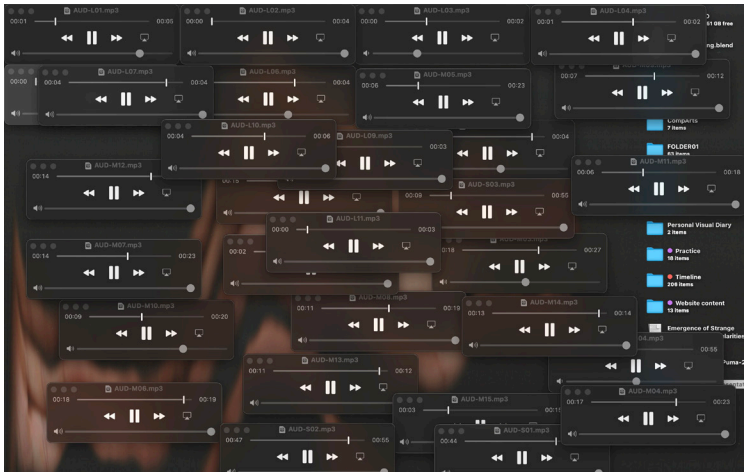
Scan of the first paper draft and the questions that shaped the personalities of the objects.

Technical Implementaiton_
Phase Two - Preparing Assets

The first step was to create image assets by collaging shapes from previous drawings I had made, and categorizing them according to their sizes in the drawings: large, medium, and small. I then began organizing the recordings in a similar way, classifying them as long, medium, and short.



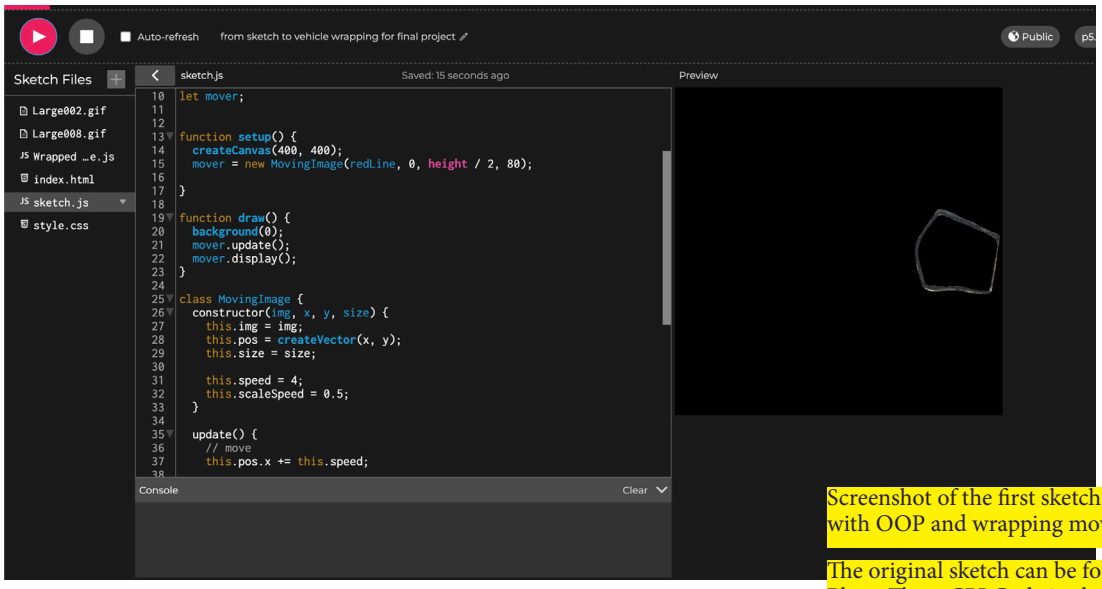
Screenshot of the layout of images collaged from drawings and organised as assets.



Screenshot of the layout of the selected audio files, aligned with the image assets.

Phase Three - Getting Back to Coding

This phase is where I began collecting code sketches and researching similar p5.js examples. I relied on The Coding Train tutorials on OOP and found sketches to reuse when creating the code for CPI. To ease back into coding after the holidays, my first step was to start with a very simple wrapping movement using an image. I focused on the preload() function to upload images and audio as a starting point, and began by creating a class for a simple object.



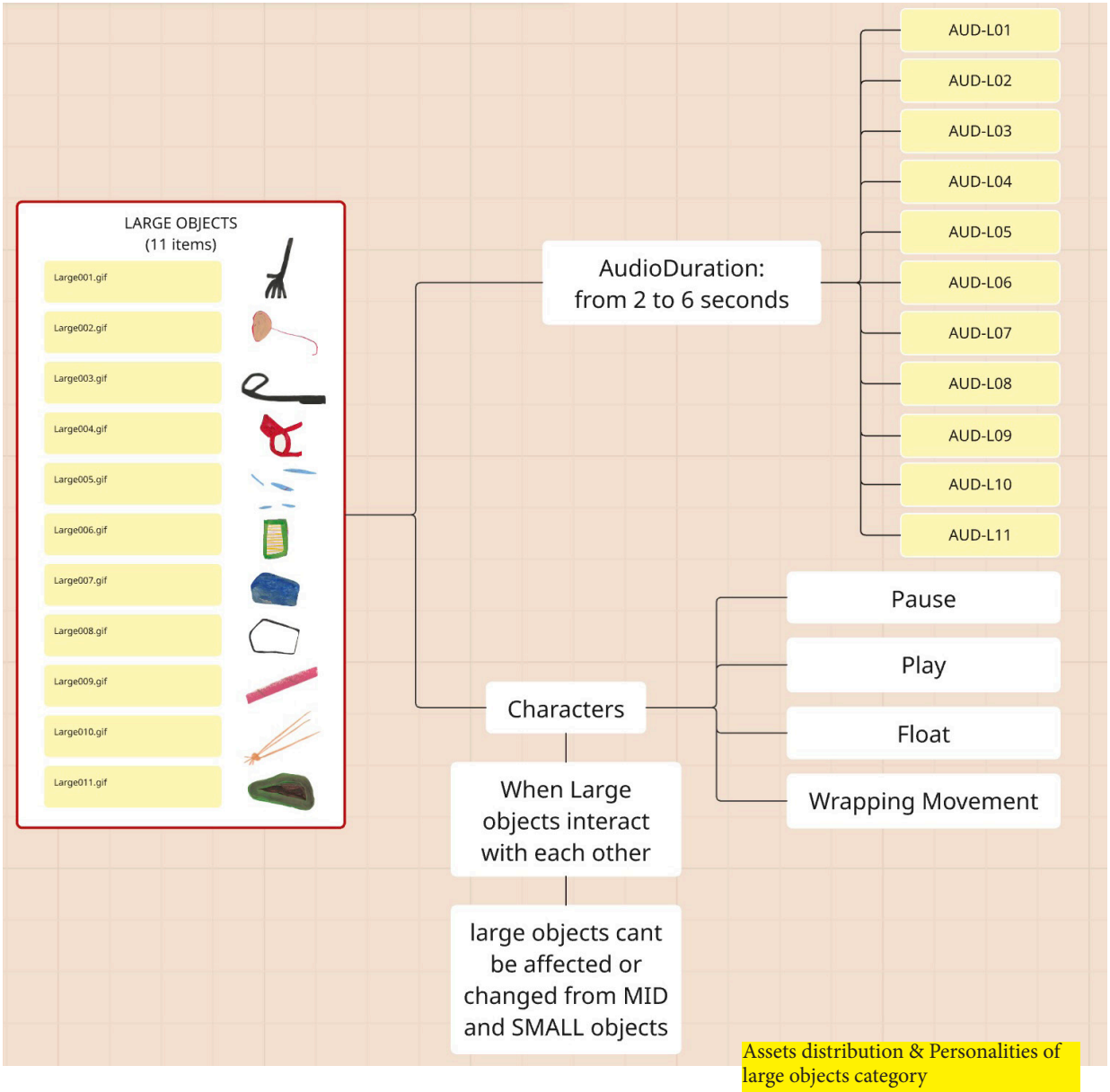
Screenshot of the first sketch experimenting with OOP and wrapping movement.

The original sketch can be found under A. Phase Three CPI Code in the coding file.

Technical Implementaiton_
Phase Four - CPI Project Map

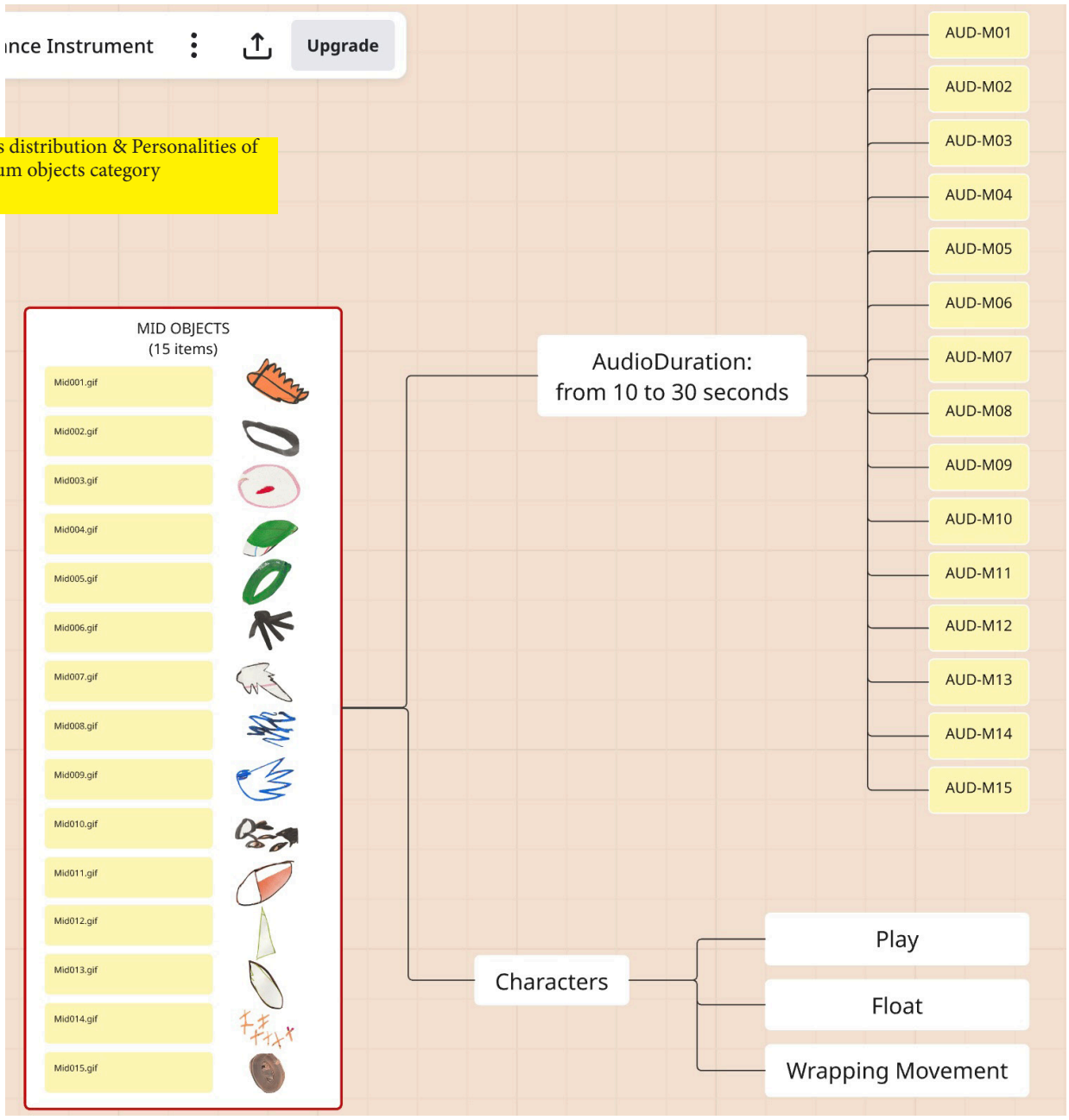
In this stage, I assigned different personalities to each category of objects. These diagrams helped translate ideas from paper notes into a functional coded program. By defining the classes for each category, CPI began to take shape as a working system. The distribution of object personalities was inspired by the scale of the collaged drawings: large objects were paired with short audio clips (under 10 seconds), medium objects with sounds around 30 seconds, and small objects with longer recordings of approximately 55 seconds. Each object is initialised differently based on its duration.

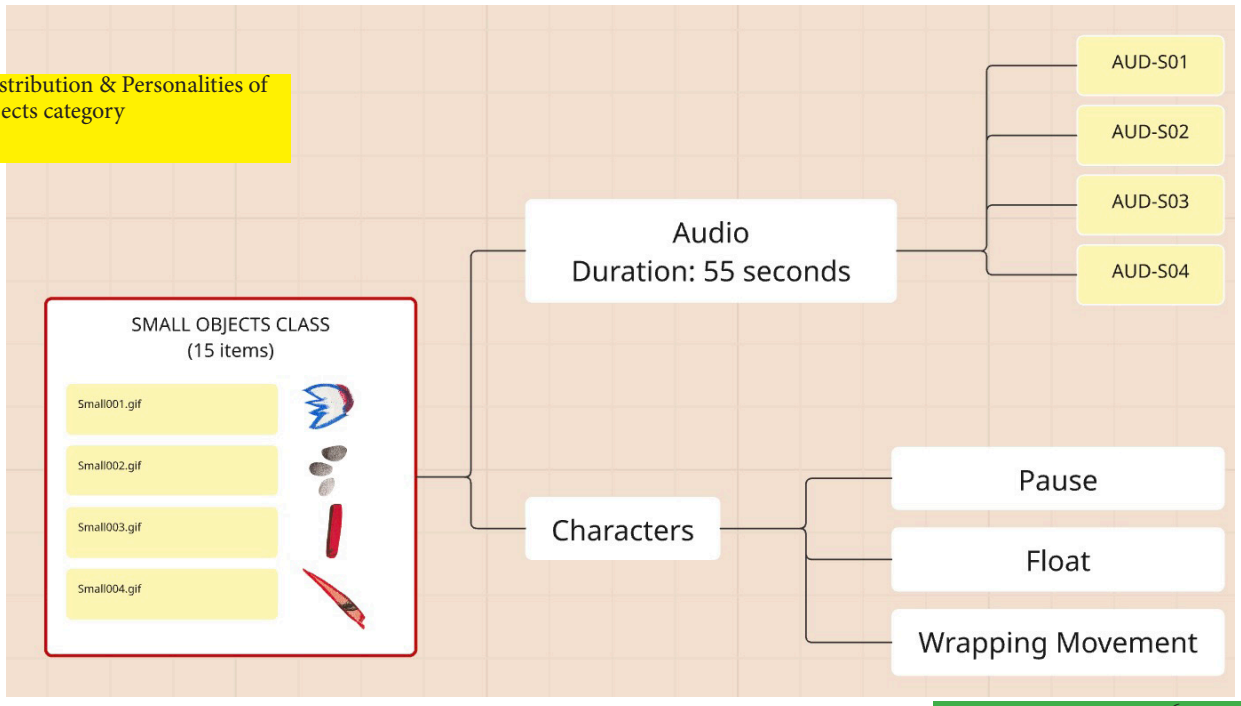
I began by testing a single object to observe its behaviour within a class structure. The object, created from collaged drawings, moves independently with slight randomness across the canvas. This sketch draws inspiration from week 8 OOP examples “Ball Class OOP” and The Coding Train tutorials, replacing basic shapes with preloaded images using the preload() function.



Assets distribution & Personalities of large objects category

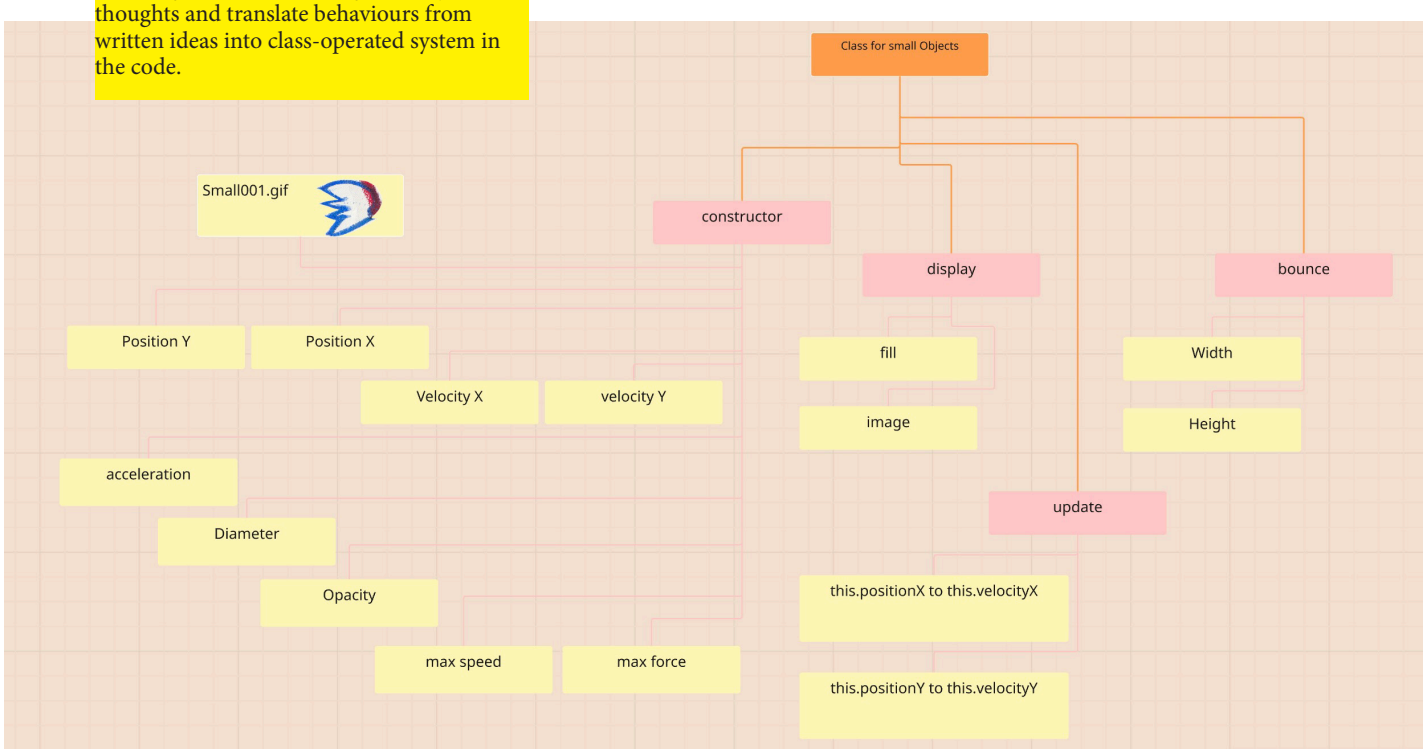
Technical Implementaiton_
Phase Four - CPI Project Map

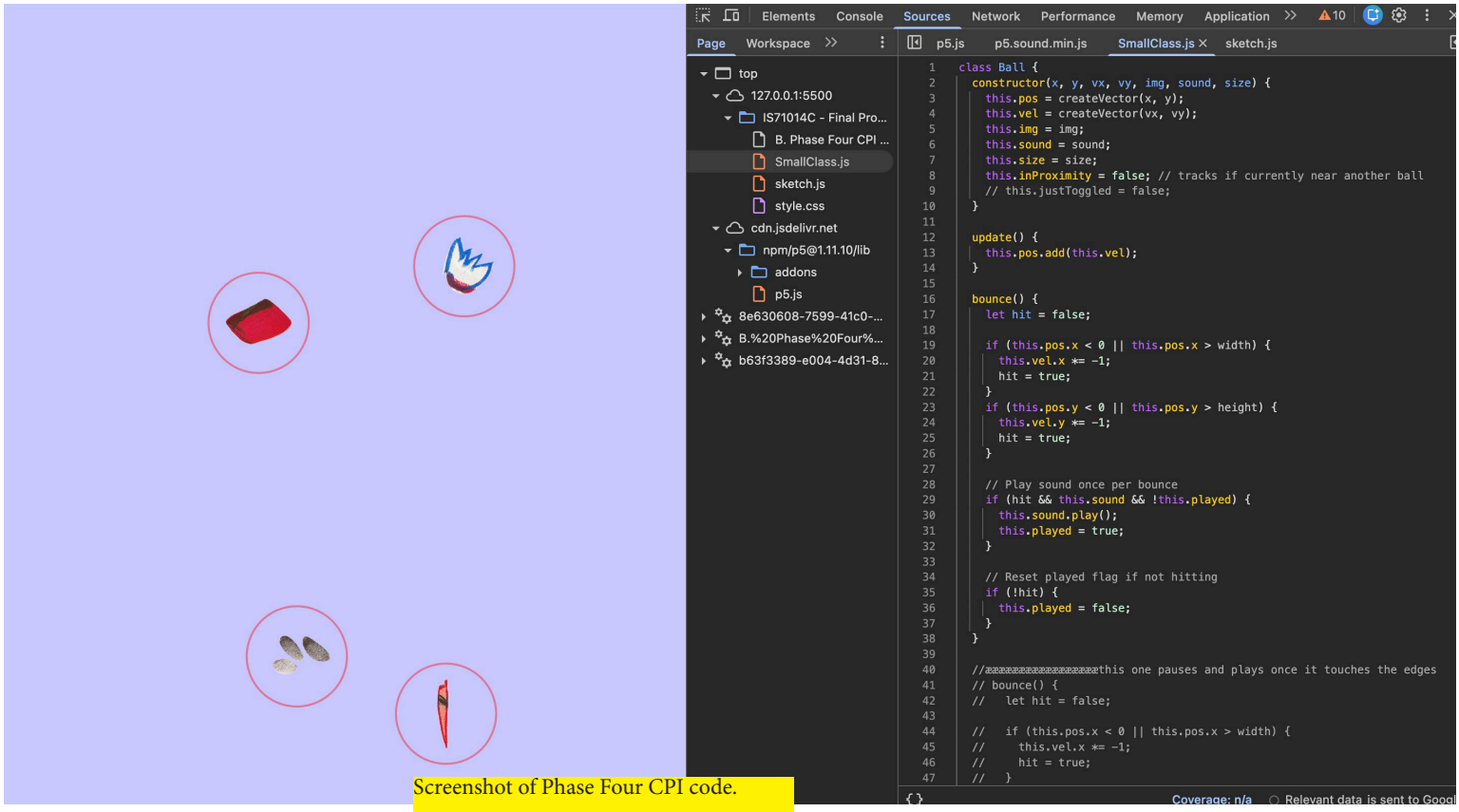




Technical Implementaiton_
Phase Four - CPI Project Map

This diagram helped me organise my thoughts and translate behaviours from written ideas into class-operated system in the code.



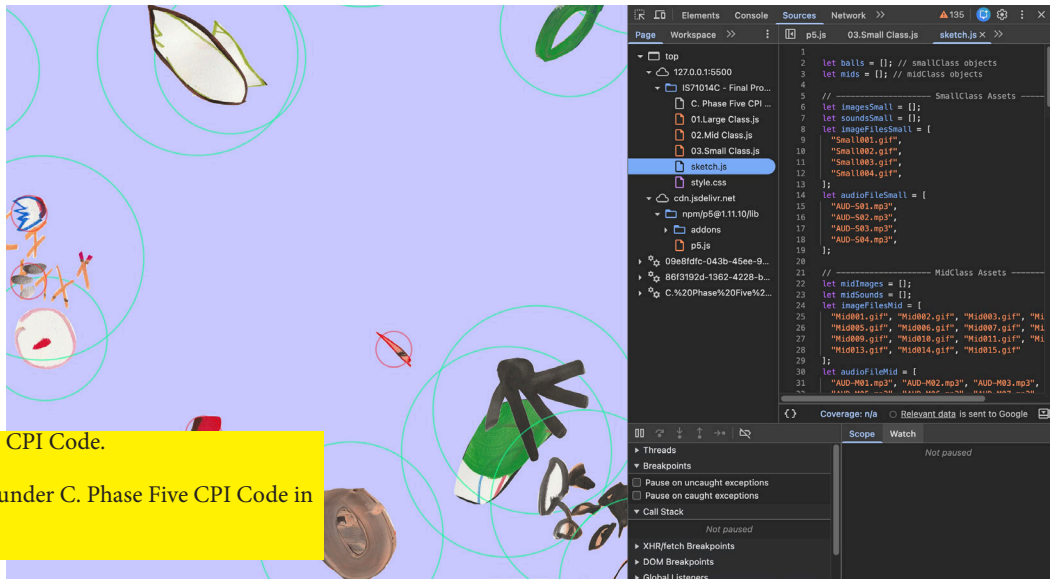


Screenshot of Phase Four CPI code.

The original sketch can be found under B. Phase Four CPI Code in the coding file.

Technical Implementaiton_
Phase Five - Reconfiguration

At this point, I realised that I could not implement all three categories of audio and interactions within a single code sketch, as the original plan was too complex. This led me to reconfigure the system into a simpler and more manageable implementation.
After completing the first code sketch, I realised I needed to take a step back and rethink the personalities. I rearranged the classes and changed the dynamics of the categories, and the diagram was optimised for better performance.

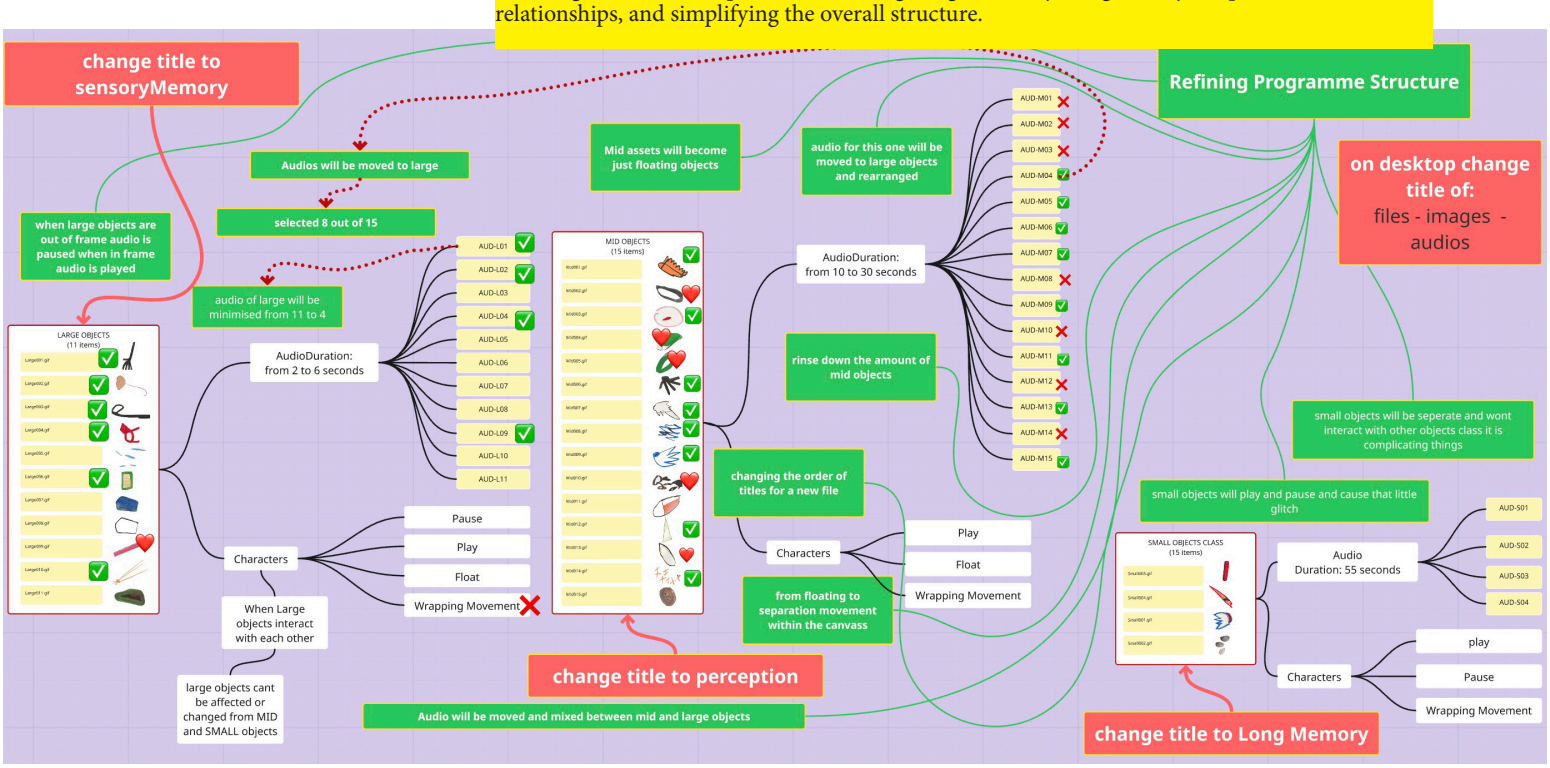


Screenshot of Phase Five CPI Code.

Find the original sketch under C. Phase Five CPI Code in the coding file.

The class titles shifted from size-based labels to names that reflect what they actually represent. Since the audio recordings come from the past, I decided to name the classes according to the length and type of memories they evoke for me. I then simplified the audio interaction, as it had become too complex for me to understand and debug the code. At this point, I received assistance from ChatGPT to support the debugging process and to help develop other aspects of the system, which are annotated and referenced within the code.
Large Objects = Sensory Memory >13 audio recordings
Medium Objects = Perception > visual-only, independent objects floating on the canvas
Small Objects = Long Memory > 55-second audio recordings

This diagram shows the process of reconfiguring and readjusting the object's personalities, audio relationships, and simplifying the overall structure.

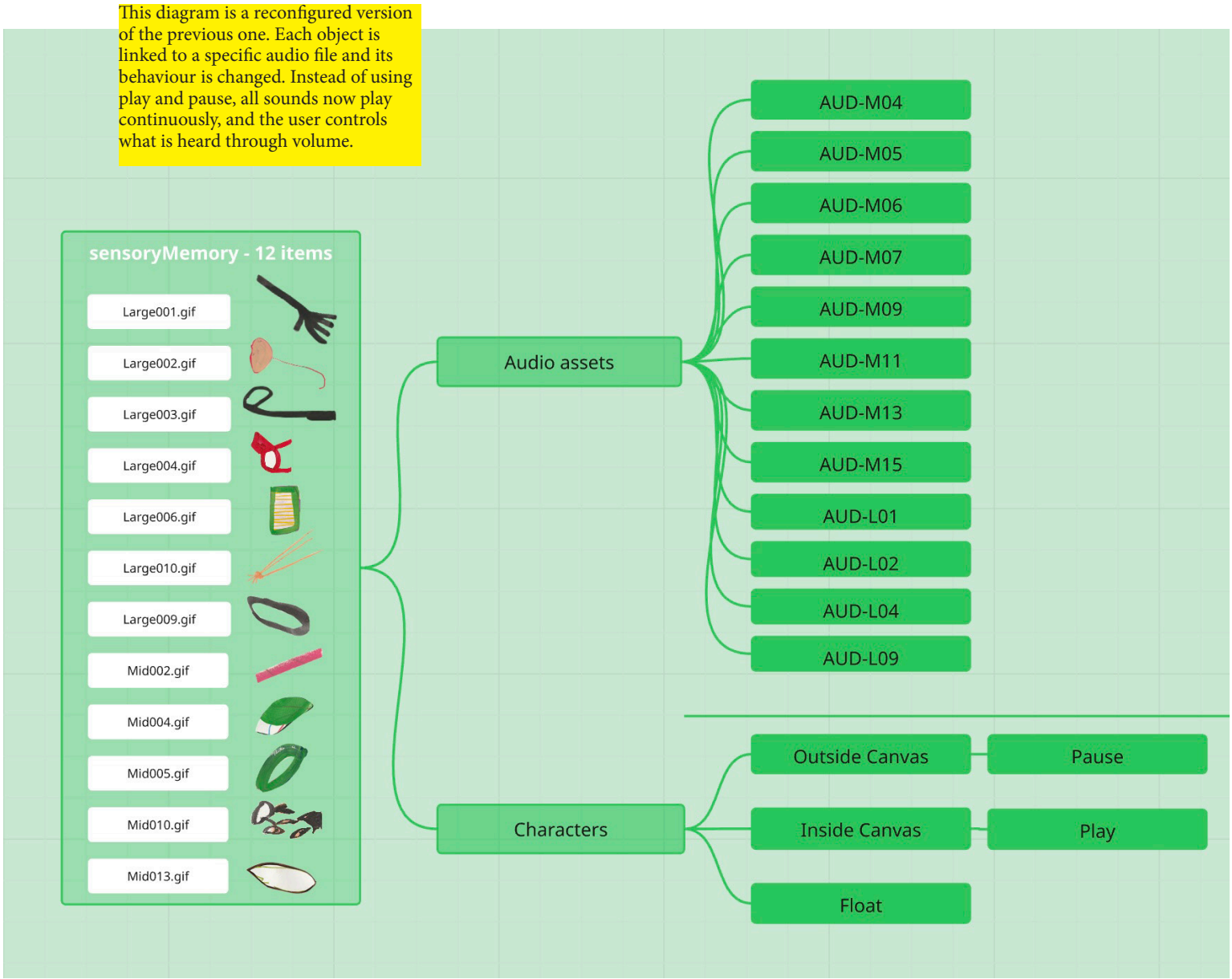


Technical Implementaiton_
Phase Six - Instrument Assembly

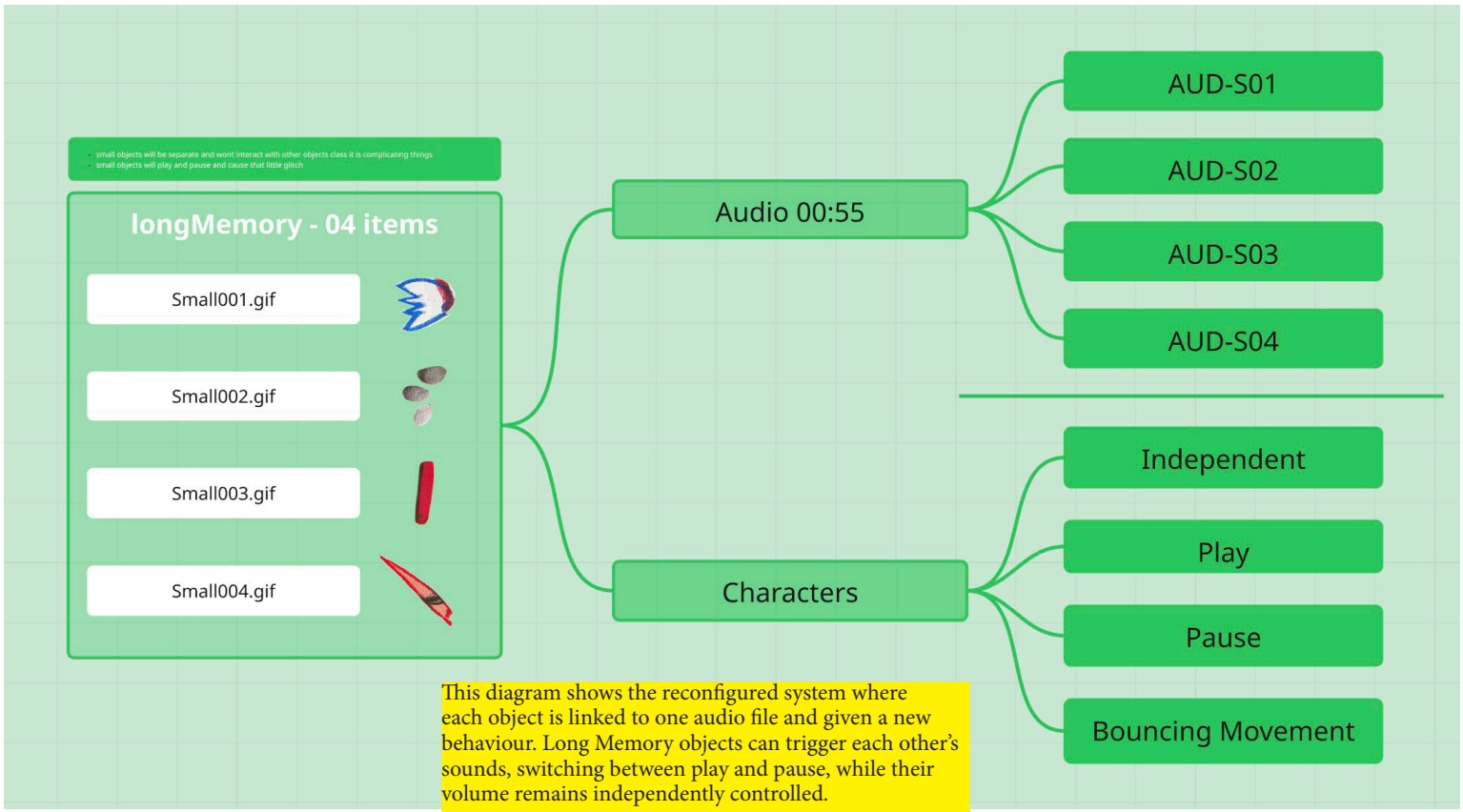
This was the final and most recent part of the code's evolution. After simplifying the personalities and minimising their interactive proximity with one another, I began focusing on the user interface and interaction. This phase was crucial, as it represents the meeting point of the project's objective: to create an instrument that is independent, yet open to interaction and improvisation.

I could have left the drawings annotated to show which audio and image I was controlling, but that would not reflect the chaos I value in my performances. Instead, I chose to keep the system mysterious and unknown.

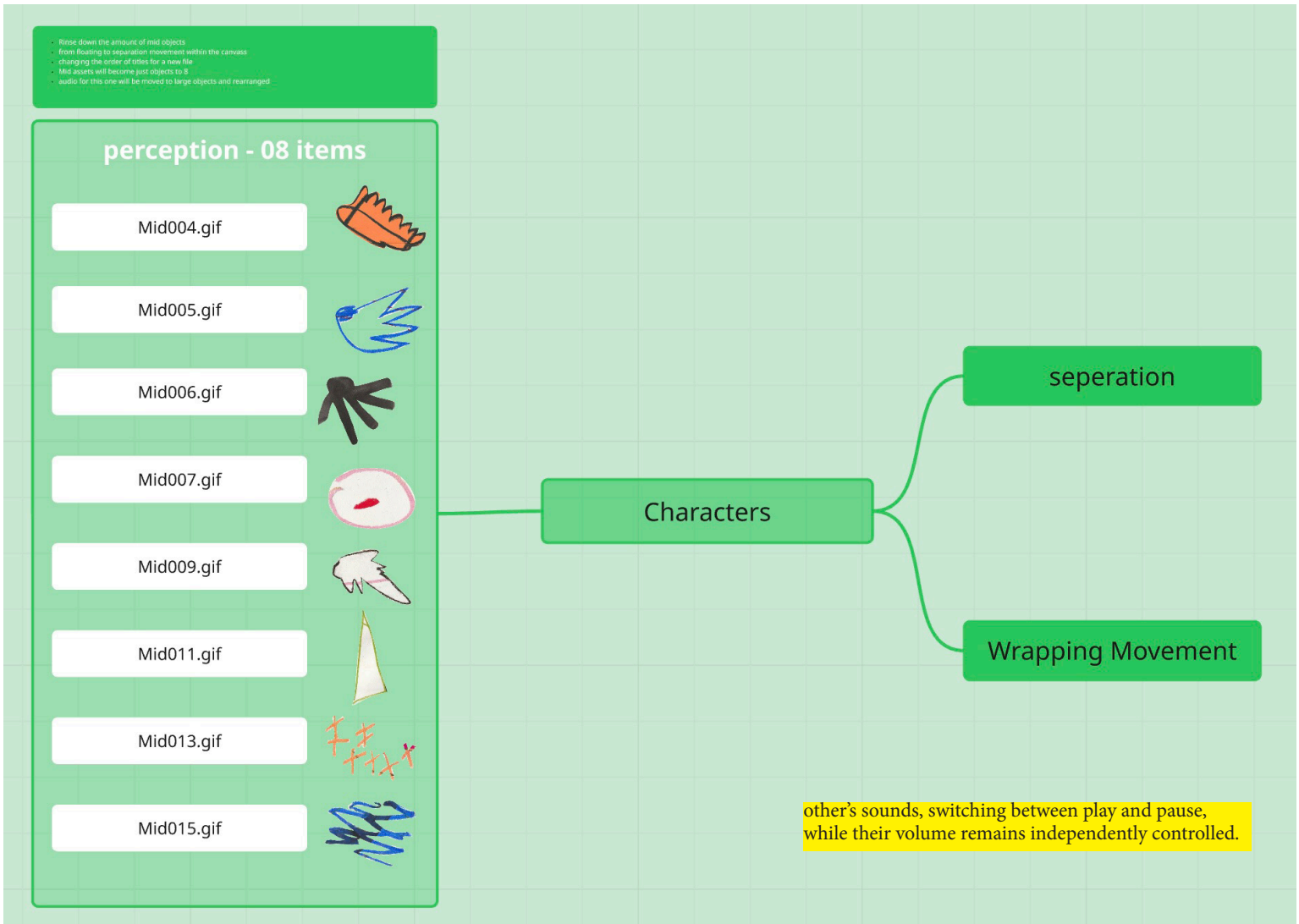
In here, I relied on ChatGPT to help me implement the sliders and to assist with adding a circular zone in the centre of the canvas that applies reverb and reverses any sound that passes within its radius.



Technical Implementaiton_
Phase Six - Instrument Assembly

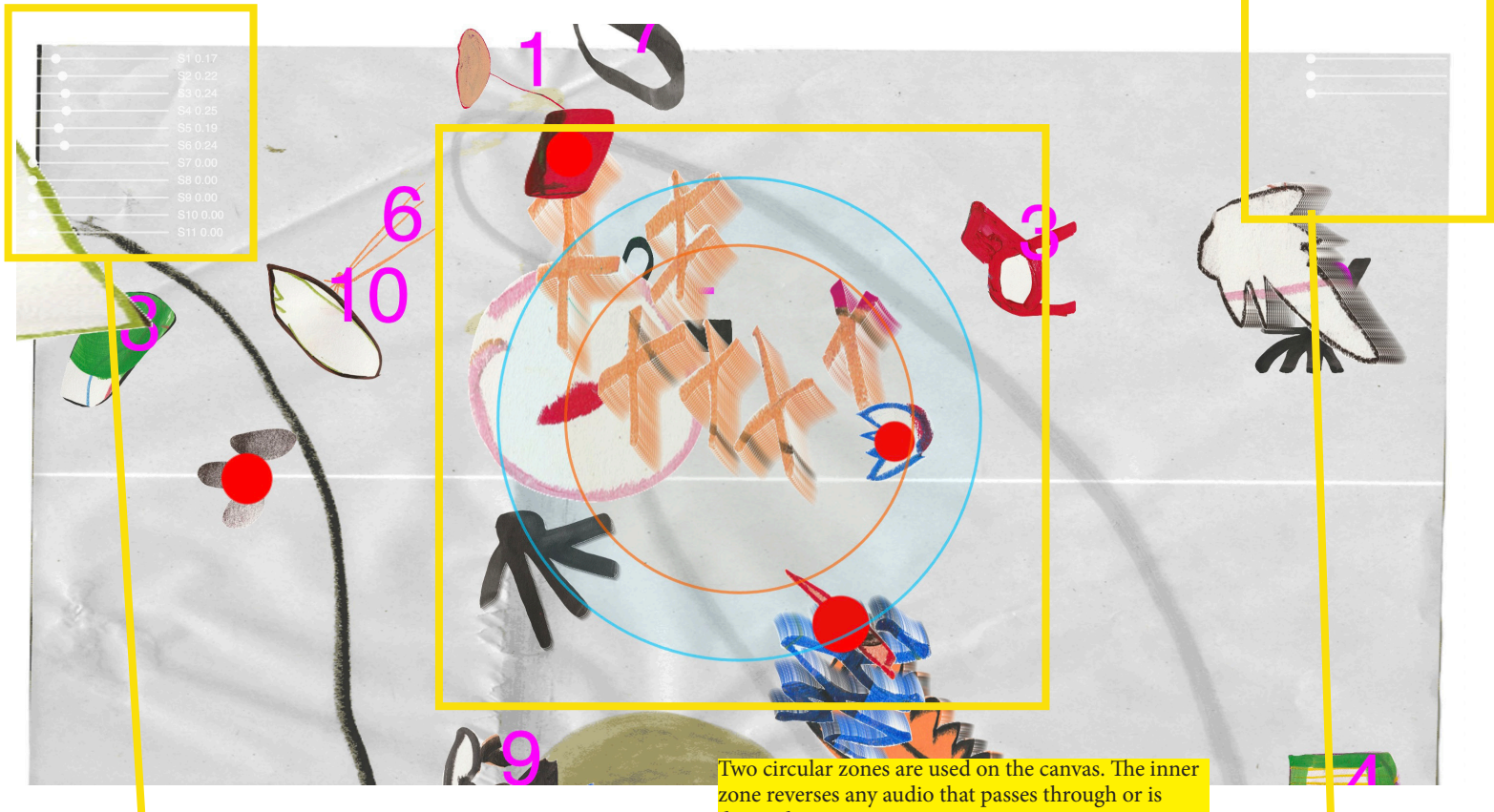


This diagram shows the reconfigured system where each object is linked to one audio file and given a new behaviour. Long Memory objects can trigger each other's sounds, switching between play and pause, while their volume remains independently controlled.

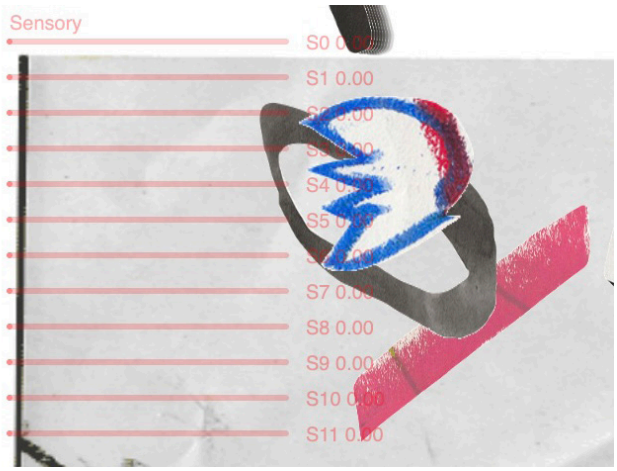


other's sounds, switching between play and pause, while their volume remains independently controlled.

Technical Implementaiton_
Phase Six - Instrument Assembly

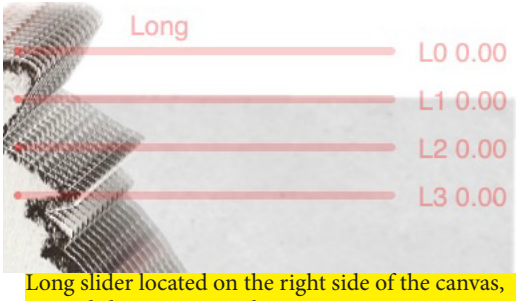


Two circular zones are used on the canvas. The inner zone reverses any audio that passes through or is dragged into it.



Sensory slider located on the right side of the canvas, controls sensoryMemeory objects

Screenshot showing the volume slider system used in CPI. Each slider controls the volume of a single sound object. The sliders are drawn directly on the canvas and mapped to individual audio files, allowing the user to adjust each sound independently during the performance.



Long slider located on the right side of the canvas, controls longMemory objects

Reflection & Future Development_

Configured Performance Instrument code, it has become an interactive, time-based audiovisual system. that translates an improvised performance into code. The instrument consists of multiple autonomous visual objects, each paired with a unique sound recording that represents a fragment of memory collected at different times and locations. These objects move independently across the canvas, looping their audio continuously. The user can influence the system by adjusting individual volume sliders and by moving objects through spatial zones that apply reverb or reverse their sound buffers. CPI functions as a semi-autonomous instrument: responsive to interaction, yet driven by chance, coincidence, and temporal layering rather than deterministic control.