

Sound Drawing Tool - Project Reflection

Assignment 2D Individual Project Reflection

1. PROJECT OVERVIEW

My project is a synesthetic audiovisual instrument called the Sound Drawing Tool. The most creative part is how drawing gestures directly create music - horizontal position maps to pitch, and when you release the mouse, all your strokes play back as a synchronized symphony. You can switch between piano, violin, and guitar modes, each with unique colors and sound characters. It turns drawing into genuine musical performance where visual and sonic elements are inseparable.

2. FIRST REFLECTION: SYNCHRONIZED PLAYBACK

Recount

The synchronized playback was the most challenging feature. I wanted all strokes to play simultaneously like an orchestra, not sequentially. I stored completed strokes in a drawingData array, with each stroke containing points that have position, color, rotation, and instrument data. During playback, I iterate through all strokes at once, triggering notes through Tone.js based on X position. I used millis() timing to keep everything synchronized at 20ms per point. The harmony system detects multiple instruments and adds complementary notes at 12% probability to enrich the sound.

React

When it first worked, it felt magical. The sound was way more musical than expected - like a real composition, not random notes. Watching the animation while hearing the music creates this satisfying feedback loop. Simple drawings make interesting melodies, and layering different instruments can create genuinely beautiful pieces. It feels like playing a real instrument.

Analyse

The success comes from key decisions. Simultaneous playback makes it feel like ensemble music rather than just sounds. The X-position-to-pitch mapping creates predictable patterns - you can learn to "play" it and anticipate sounds. Conditional harmony adds depth without muddiness. Visual feedback during playback creates clear cause-and-effect understanding.

Improve

I'd add more sophisticated harmony algorithms based on musical keys would sound more professional. Adding save/export functionality would make it feel like a proper creative tool instead of just a toy.

3. SECOND REFLECTION: REVERB AND ECHO EFFECTS

Recount

My tutor suggested adding reverb and echo effects to make the sound more spatial and interesting. I tried implementing both using Tone.js effects. I created a Reverb node with decay time around 2-3 seconds and connected it to my synths. For echo, I used FeedbackDelay with delayTime of 0.25 seconds and feedback around 0.3. The idea was that reverb would add warmth and space, while echo could create rhythmic repetition that might enhance the musical patterns.

But when I tested it, the results were disappointing. The reverb made everything sound muddy and washed out, especially when multiple instruments played together. The echo was even worse - it created this cluttered mess where you couldn't distinguish between the original notes and the delayed ones. When strokes overlapped, the echoes would pile up and it just sounded chaotic. I ended up removing both effects and keeping the dry sound.

React

Honestly, I felt frustrated giving up on the effects. I wanted them to work because they sounded good in theory - professional music uses reverb and delay all the time. It felt like I was missing something or doing it wrong. But I also felt relieved when I removed them because the clean sound actually worked better for the instrument.

Analyse

Looking back, the effects didn't work because of how my instrument generates sound. Professional music uses effects on carefully composed, spaced-out notes. My tool generates tons of notes in quick succession from multiple overlapping strokes. Heavy reverb on dense, fast notes creates mud. Echo on overlapping patterns creates chaos because the delays stack up unpredictably.

The other issue is that effects need context. A 2-second reverb tail works in a song with space between notes. But my instrument plays continuously during strokes, so reverb tails never have time to decay before new notes trigger. Same with echo - feedback loops need gaps to breathe.

Improve

If I tried effects again, I'd use them much more subtly. Very short reverb with 0.5-1 second decay could add slight warmth without mud. Or apply reverb only to specific instruments - maybe violin gets reverb for smoothness while piano stays dry for clarity. For echo, I could use much shorter delay times (0.1 seconds) with low feedback (0.1-0.2) to create subtle doubling rather than obvious repetition.

Better yet, I could make effects conditional based on drawing speed or density. Slow, sparse drawings could get more reverb and echo because there's space for tails to develop. Fast, dense drawings stay dry. Or let users toggle effects on/off with keyboard shortcuts so they can choose based on what they're creating. The key lesson is that effects need to match the musical context, not just be applied blindly.

CONCLUSION

This project taught me that good creative coding isn't just about adding features - it's about understanding why features work or don't work in context. The synchronized playback succeeded because I thought about how users would experience ensemble music. The reverb/echo failed because I applied professional audio techniques

without considering how my instrument generates sound differently. Sometimes the best decision is removing things that don't serve the experience. If I start another project, I'll test features more critically and be willing to cut what doesn't work, rather than forcing everything to fit.