

DEVELOPMENT 1 of DFAM LITE

I. Assets from prototype:

1. Oscillator 1
2. Tempo Slider
3. Start Button
4. Stop Button
5. Reset Button
6. 8 Step Sequencer
 - a) On the front end, there are 8 “steps,” each has code in HTML that allows it to be either “active” or not, meaning that depending on the step, it is on the square for the related step is yellow or it's left as a blank square.
7. Master Gain Control
8. Master Decay Control

B. Notes:

1. I only realized later that the sound needed to be initialized to start. Before, when working on the code, I would continuously hit “save” without refreshing the page, which would be enough to initialize the sound. So, if you were thinking why the sound wasn't working, that's most likely why.

II. Implementations of Development 1

1. Initialize sound button
 - a) This starts the oscillators
2. Oscillator 2
3. Oscillator type switch
4. Oscillator Freq Slider
5. VCO Decay Slider
6. VCO Amount Slider
7. Filter
8. Filter Freq Slider
9. Filter Type Switch
10. Filter Q Slider
11. Pitch Control For each Step in the Sequence
12. ADSR Nodes for each step in the Sequence

B. Notes

1. Issues:
 - a) When the user changes the Tempo while the sequencer plays, the sequence stops momentarily while the slider changes value. I believe this is because the function for changing the tempo, the sequencer, has to be cleared before it can reinitialize the sequencer with a new tempo. So, as of now, I haven't found a way around this, but I want to because that's one of the central parts of the fun when playing with the actual DFAM: being able to “seamlessly” change the tempo while it plays.

- b) If the user hits the start button more than once, a second sequence that runs in tandem begins. I plan on replacing the start and stop buttons with just one button that, if the sequencer isn't running, will start the sequencer and, if it is running, will stop it. This way, there's no way to start the sequencer more than once.

2. Successes:

- a) After a LONG time, I figured out how to get the decay working as I wanted it to. In the prototype, I just had one instance of an ADSR Node that would increase and decrease the gain for every step. This had problems because if the decay of the sound were longer than the interval time of the sequencer, then it would almost take the difference in time and turn that into decay (amongst other problems aswell). So, to work around this, I made 8 different ADSR nodes specific for each step in the sequencer. That way, I could increase and decrease the decay without worrying about how it would be affected when another starts.
- b) Using the outline I created with the separate ADSR Nodes, I was able to change other values based on a specific step in the sequence, so creating pitch sliders for each step wasn't too hard. This adds a lot of character to the code's functionality.

3. Plans for the next development:

- a) White Noise generator:
 - (1) I want to implement a white noise generator to get closer to the sound of the original DFAM. This will help with some of the "snare" sounds it can make.
- b) Mixer for each of the Oscillators and the White noise generator
 - (1) For this, I would have to reconfigure how the nodes connect; I think the oscillators will connect to their specific gain nodes and then have those connect with the ADSR Nodes. This will take some trial and error.
- c) Record and Export functions.
- d) VCF function
 - (1) Something that will change the value of the Filter each time the sequencer makes a step to bring it up to a specific value and then back down to the original value based on a decay value.