Bao’ s super synthesizer

The assignment 3

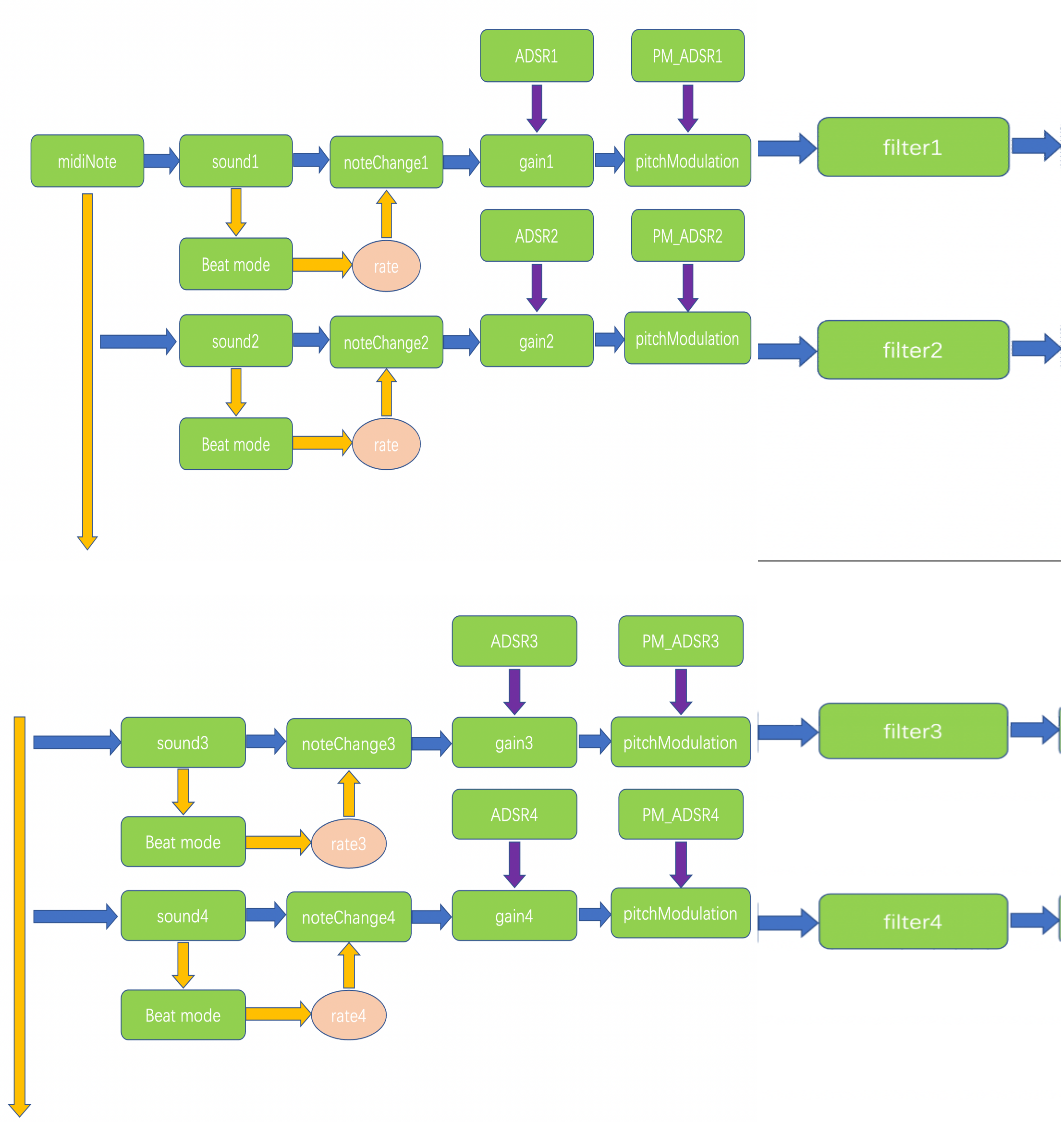
Intro:

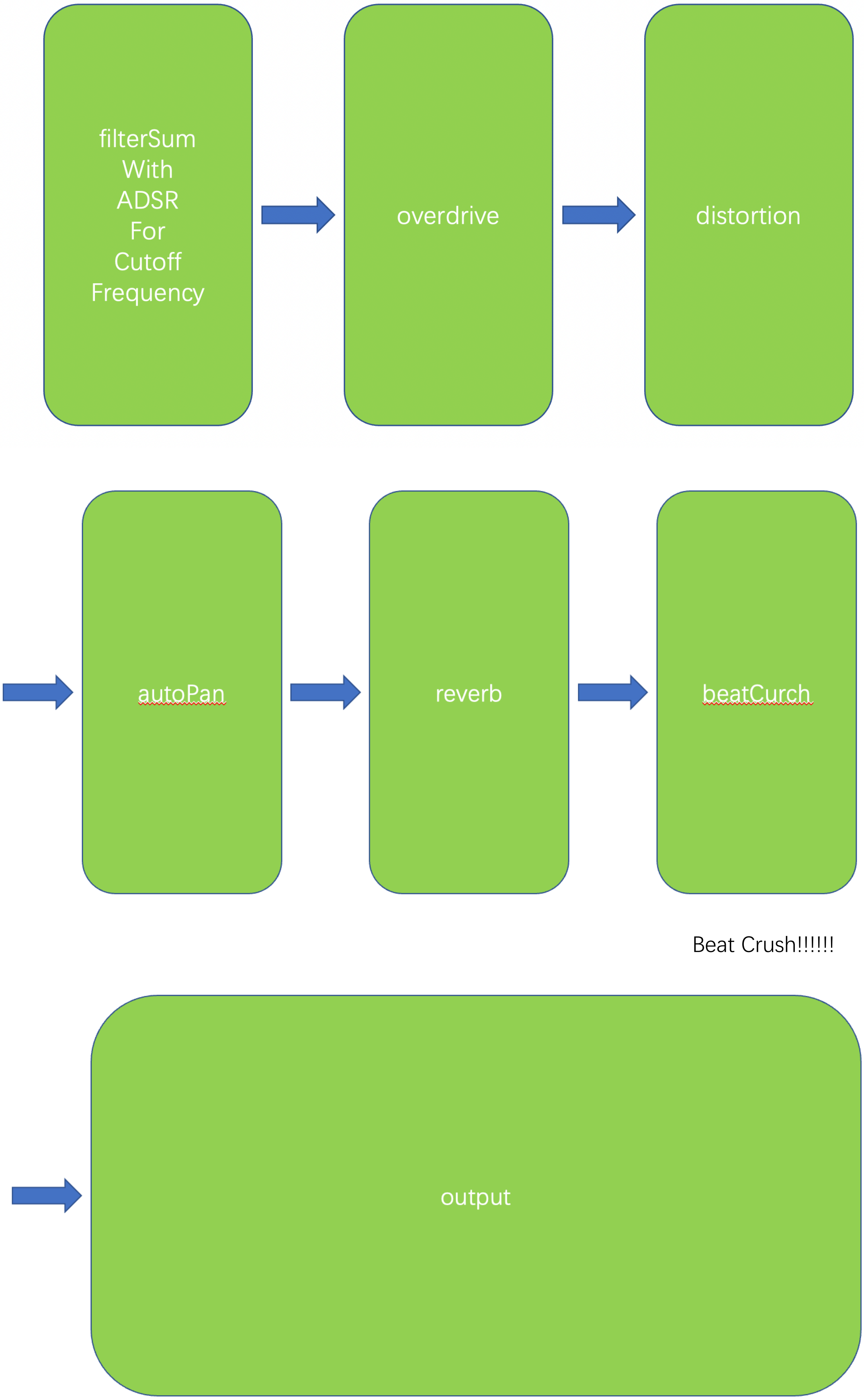
This time I try to use my knowledge to make a very cool and complicated synthesizer. I named this project Bao’ s super synthesizer, because this synthesizer has a lot of useful and creative functions, which can allow you to make some very amazing sounds. For me, I will be sure to use this synthesizer in my future music production.

Description:

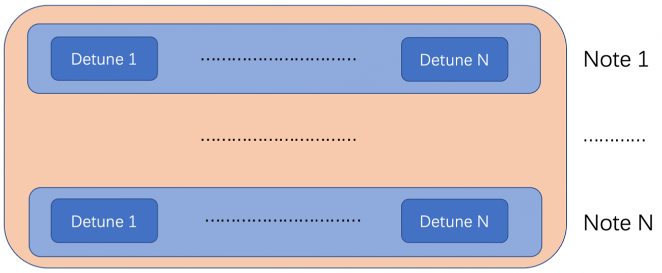
This project can be mainly divided to several different parts: **sounds, beat mode, note change, gain modulation, pitch modulation, filter for each sound, filter with ADSR for the cutoff frequency, overdrive, distortion, auto pan, reverb, and beat crush.**

The whole flow chart is as follows：

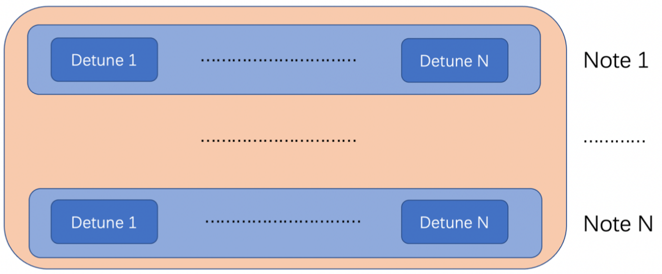



filterSum
With
ADSR
For
Cutoff
Frequency
overdrive
distortion
autoPan
reverb
beatCurch
Beat Crush!!!!!!
output

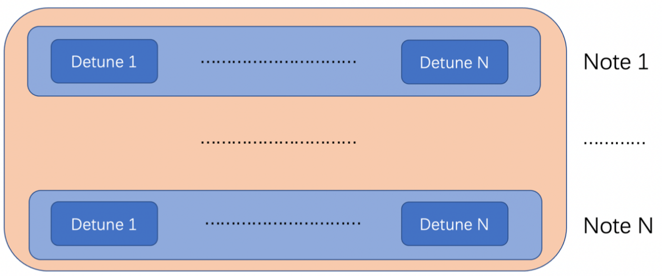
The sounds module inside is like the flow chart as follows:



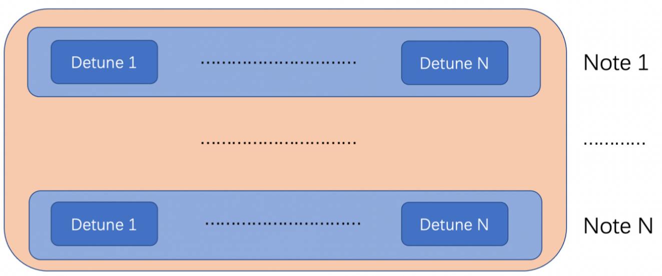
No.1



No.2



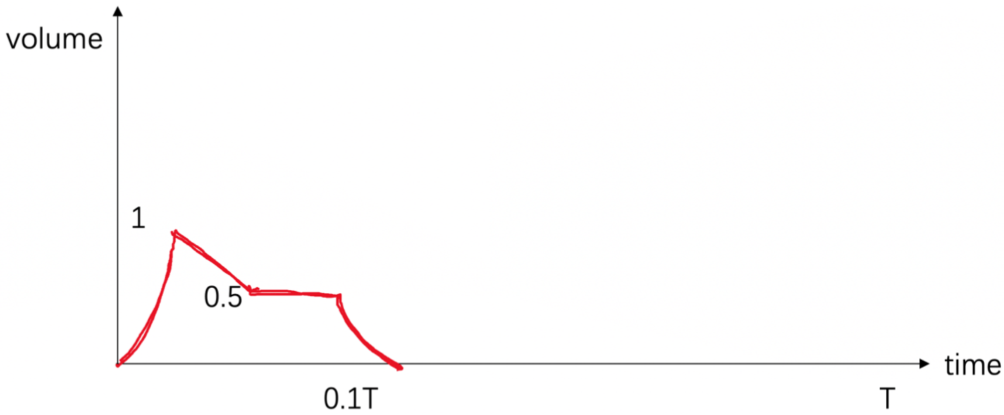
No.3



No.4

Firstly, for the sounds module, I fixed DetuneN, and the max size of the vector to 8, and NoteN to 4, and I make 4 sound groups like the picture inside the sounds module. Each group contains several different notes, which the quantity can be set. Inside the group, each notes can have many sub-oscillators with detunes whose frequency and quantity can be set as well. What' s more, each sound group can have four types, which mainly contains sin, square, triangle, and saw. Therefore, these four groups can make a very rich sound. After this, I also add a little white noise in order to make the sound warmer.

Then, the beat sound module. This time, I use an artificial ADSR for the sound so that the sound will not be continuous, instead, the sound will be non-continuous.



The picture shows how my ADSR works through a whole T. I create an envelope class which inherits from the phasor class. From the picture, we can see that during 0.1T to T, the sound will be nothing, so in this way, the sound will be like a heartbeat. I set 4 beat rate parameters for each sound, so you can use them to make very interesting sounds.

Then, I add a note change mode to my synthesizer. In this module, you can change note for each sound, perhaps a fifth chord or a third chord by changing the parameters. If you set the value 0, it means that the note will be the original note and will not be changed.

Then, I add an ADSR envelope for each sound’ s amplitude. Also, I set a basic gain for each sound. Therefore, you can use this module to make a modulation of each sound’ s amplitude and make the sound more dynamic.

Next, I set a pitch modulation mode for each sound, and I also add an ADSR envelope for each sound’ s frequency. With this mode, you can make very rich and interesting sounds like drop frequency sounds.

Next, I add a filter for each sound, in which the cut off frequency can be set. By doing this, you can make sounds much more creative, especially with automation.

Then, I add another filter for the whole sound. Inside this filter, it is quite different that the cut off frequency can be modulated by an ADSR envelope. This function is quite powerful, because you can use this envelope to create many different sounds, such as kicks, drops, and so on.

Then, I add an overdrive mode and a distortion mode to the whole sound. I made an overdrive class, which contains basic functions of setting and processing. In the processing function, if the sample multiplies the pre-gain is beyond threshold, the sample will be lower down to some extent. What’ s more, I create a distortion class which basically inherits from the overdrive class. The processing function was changed. If the sample is beyond threshold, this time it will be totally cut down and the sample will be nearly equal to the threshold. In this function, I used the tanh function make the process smoother. In this mode, the threshold, the overdrive ration, and the pre-gain can all be set with you in real time.

Next, I make a pan class with stereo output. I use a sin wave LFO to modulate
the panning, so it will move from side to side. Also, I set the width of the auto pan. This means that if the width is small, the sound will be more likely to appear in the central area, but if the width is large, the sound will be more likely to appear in the two sides. Also, the LFO rate and the width can be both set with yourself in real time.

Next, I add a reverb to make the whole sound milder. Inside this reverb function, I set all the parameters include dry value, wet value, width, room size, and so on. All the parameter can be set in real time.

Last, and the best part in this project I believe, is the beat crush mode. In this mode, I use a unique way to simulate down sample rate. Samples will be the same in each X times, which can simulate down sample rate. X can be two, four, eight, sixteen, or any number else. In this way, the sample rate is like being divided by X. This function is very powerful because you can use this and the filter to create a kind of chiptune.

Future work:
According to the request of this assignment, I haven't made anything about the UI interface so that the interface looked very mess, so in the future, I will learn how to make good UI interface by myself and then make my synthesizer's interface neater. What's more, I want to add more functions into my synthesizer to make it more powerful and perhaps I will share this original synthesizer to my friends and classmates.

Problems I met:
Firstly, I met some troubles in understanding the constructor for the whole plugin processor. After figured out, I got in trouble with editing so many parameters in more than one thousand lines' code. This really take me a lot of time. Finally, I try to figure out a way to make beat crush. This is a hard challenge for me at that time as well.