

Project 2d heb ik gebruikt om een begin te maken om JUCE te leren. Ik heb het opgesplitst in 2 delen.

Ik ben eerst begonnen met een proof of technology midi Controller voor de Roland Alpha Juno serie. Daarna heb ik vervolgens alles wat ik aan JUCE had geleerd gebruikt om een stuk sneller een Ladder filter VST plugin te maken. Ik heb dit blok vooral gebruikt om iets nieuws te leren en niet zo zeer voor portfolio materiaal.

## **Roland alpha Juno controller**

### **Waarom:**

De Roland alpha Juno heeft geen knob per function control. Deze synth word digitaal aangestuurd wat inhoudt dat je menu diving moet doen om parameters aan te passen. Ik wou de knob per function control terug.

### **Waarom als vst;**

Deze controller is eigenlijk een vervolg op een maxpatch die ik al eerder had gemaakt. Ik wou kijken of ik van die max patch een werkende vst versie kon maken die ongeveer hetzelfde kan. Omdat die dan te gebruiken is binnen bijna elke DAW wat je meer mogelijkheden geeft voor sound design, automations etc.

### **Voor wie:**

Ik heb deze proof of technologie vooral voor mezelf gemaakt. Hierbij kon ik een introductie doen met JUCE en kijken of mijn idee mogelijk was. Als ik het project af zou maken is het gericht op mensen die een Roland alpha Juno hebben en een DAW workflow.

### **werkproces:**

Ben begonnen met de setup van mijn werk omgeving. Ik wou met mijn eigen code editor werken maar de projucer(app om projecten mee te generen) van JUCE ondersteunde alleen Xcode projecten dus moest ik werken met cmake. Vervolgens heb ik uitgezocht hoe ik sliders kon maken in JUCE.

Het moest een Midi effect worden. Dat zorgde nogal voor problemen want dit word niet ondersteund door iedere plugin extensie. Ook kon ableton mijn vst3 niet zien als ik die bouwde het maakte niet uit of het een geluids effect of een midi effect of een synth was. Door de oplossing die ik uiteindelijk heb gevonden moest ik toch wisselen naar Xcode dit was omdat ik een vst3 moest bouwen voor "any Mac". Want het werkte niet lekker als m1 only. Uiteindelijk werkte het nog steeds niet in ableton als midi effect en heb ik reaper gebruikt als test omgeving.

Vervolgens ben ik nog lang geworsteld met de threads binnen JUCE want ik wou vanaf message thread naar een MIDI thread. Dit is uiteindelijk gelukt met een nieuw MIDIOUT object in mn class.

Uiteindelijk wou ik ook nog dat de incoming sysexmessage on patch change van de synth de software slider zou kunnen aanpassen naar de juiste waarde. Dit moest met een bus tussen threads. Dit heb ik uiteindelijk niet meer geïmplementeerd want de 3 weken waren over en ik wou eigenlijk nog met iets nieuws beginnen.

### **Reflectie:**

Ik ben met dit project software matig minder ver gekomen dan ik had gehoopt. Maar ik heb wel heel veel nieuwe dingen geleerd over c++ en hoe je moet werken met referenties en frameworks. Deze dingen kon ik heel goed gebruiken bij mijn volgende project om zo erg snel een LadderFilter VST te maken.

## **Moog ladder filter:**

### **Waarom:**

Dit project was ook meer een leerdoel dan portfolio materiaal. Ik heb dit project gedaan omdat ik graag nog verder wou met JUCE en omdat ik al een tijd een Ladder Filter wou bouwen. Ook omdat ik dan wat ik in de DSP lessen van Pieter had geleerd kon gebruiken in de "praktijk".

### **Voor wie:**

Voor mij of ladderfilter liefhebbers.

### **Werkproces:**

Ik ben begonnen met het lezen van Oscillator and Filter Algorithms for Virtual Analog Synthesis Article in Computer Music Journal · June 2006. Door Vesa Välimäki en Antti Huovilainen. Daar uit heb ik de blokdiagrammen en de coëfficiënten gehaald. Vervolgens heb ik de onepole gemaakt in max. En daarna heb ik hem getest met m'n effect base class van vorig blok. Toen heb ik een nieuw JUCE project gestart deze keer meteen in Xcode. Sliders toegevoegd en gezorgd dat ik vanuit de "message thread"(thread waar in de sliders runnen) kon communiceren met de audio thread. Dit zorgde er voor dat ik de parameters van mn effect kon aanpassen. Ook heb ik xml support toegevoegd zodat als je de plugin sluit hij de waardes onthoudt. Daardoor kan je ook werken met automation binnen je DAW.

### **Reflectie:**

Ik heb nu eigenlijk een clean 4pole lowpass LadderFilter. Volgens het blokdiagram moet er voor dat het in komende signaal door een filter gaat nog een tanH waveshaper overheen om de analoge distortion van het origineel te emuleren dit heb ik nog niet toegevoegd omdat ik niet genoeg tijd had om een oversampler te bouwen. Ook wil ik nog de oberheim Xpander trick toepassen. En de code die er nu is wat opschonen dingen uit de baseclass halen die ik niet gebruik. waardoor je het aantal polen en het soort filter kan aanpassen in runtime. Dit zijn nog dingen die ik zou willen toevoegen. Bij dit project heb ik geleerd over audio processing in JUCE en DSP.

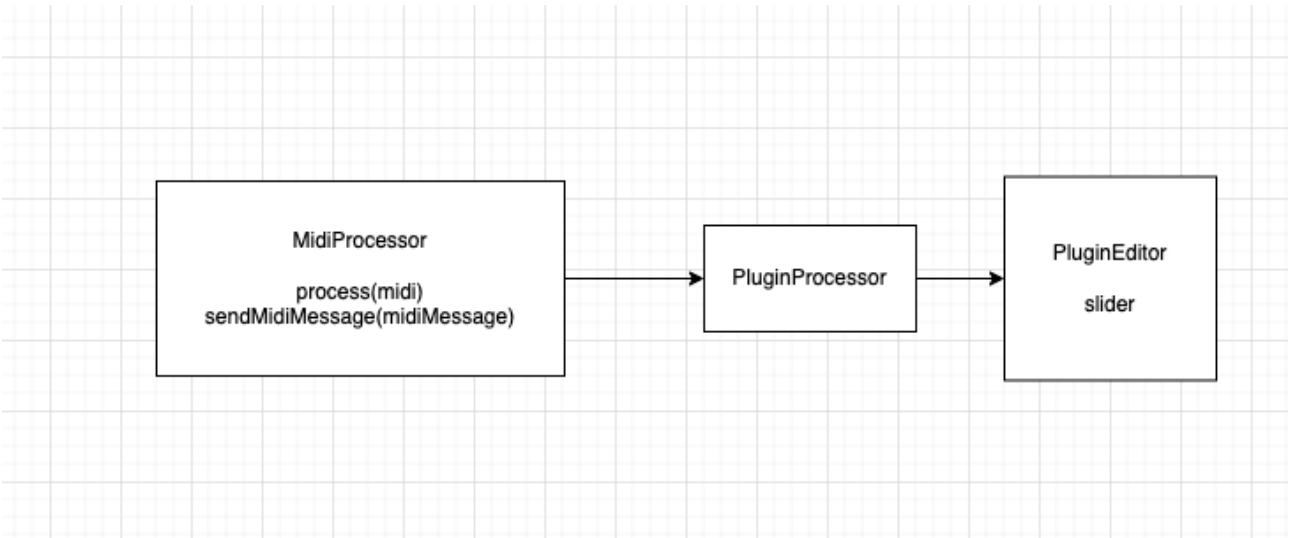
### **Kort stukje game audio.**

Ik heb dit blok als 3de vak SID gevolgd en heb daar gewerkt met Wwise ik ben tot de conclusie gekomen dat dit programma niet echt interessant is voor de CSD lessen omdat alles er eigenlijk al in zit. Het zou wel toepasbaar zijn bij KO want je hebt mogelijkheden om Wwise te gebruiken als een soort performance tool. Er is wel een mogelijkheid om plugins voor te schrijven maar de meeste nuttige audio effecten zitten er al in. Maar game audio en plugins voor fmod/Wwise kunnen wel erg interessant zijn. We hebben het bij Jorrits les gehad over genereren van natuurgeluiden met noise dit zou je natuurlijk prima als plugin voor in een game kunnen maken en zo bijvoorbeeld adaptieve regen kunnen genereren.

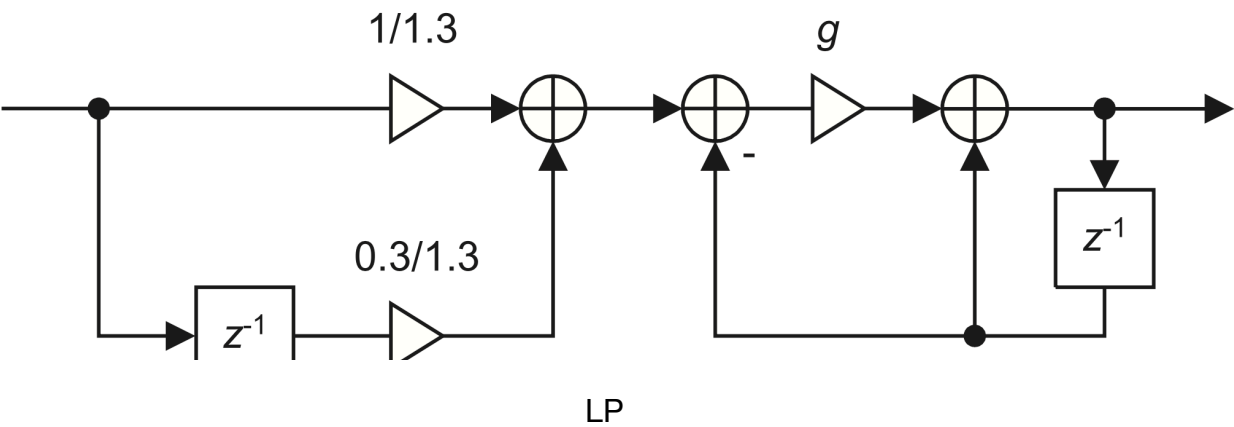
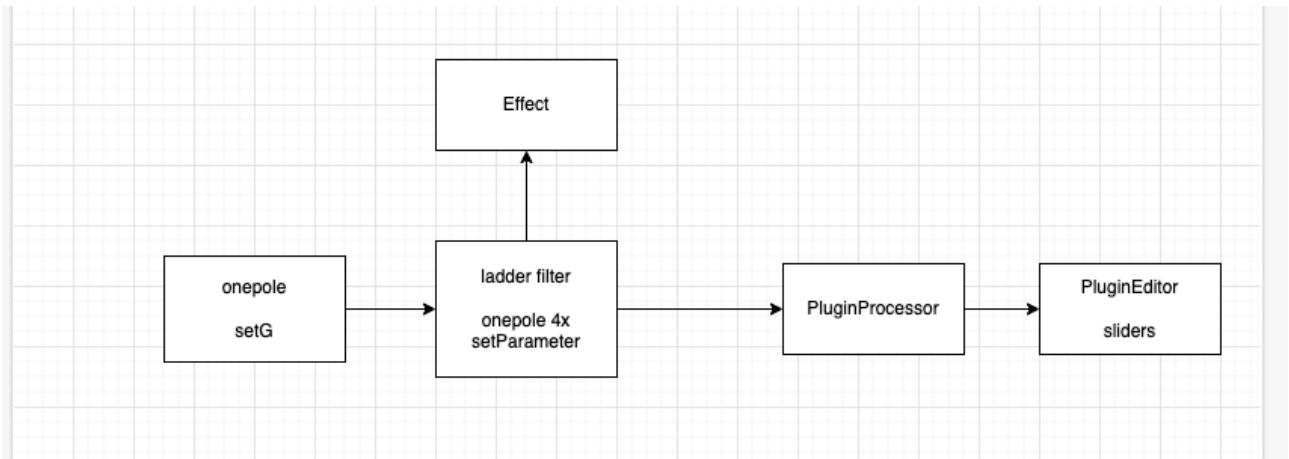
Wwise vs Fmod. Persoonlijk vind ik fmod fijner werken omdat je daarin toch net iets meer de DAW workflow hebt waar ik aan gewend ben. Fmod runt ook native op macOS en Wwise runt via WINE wat wel werkt maar niet echt optimaal is.

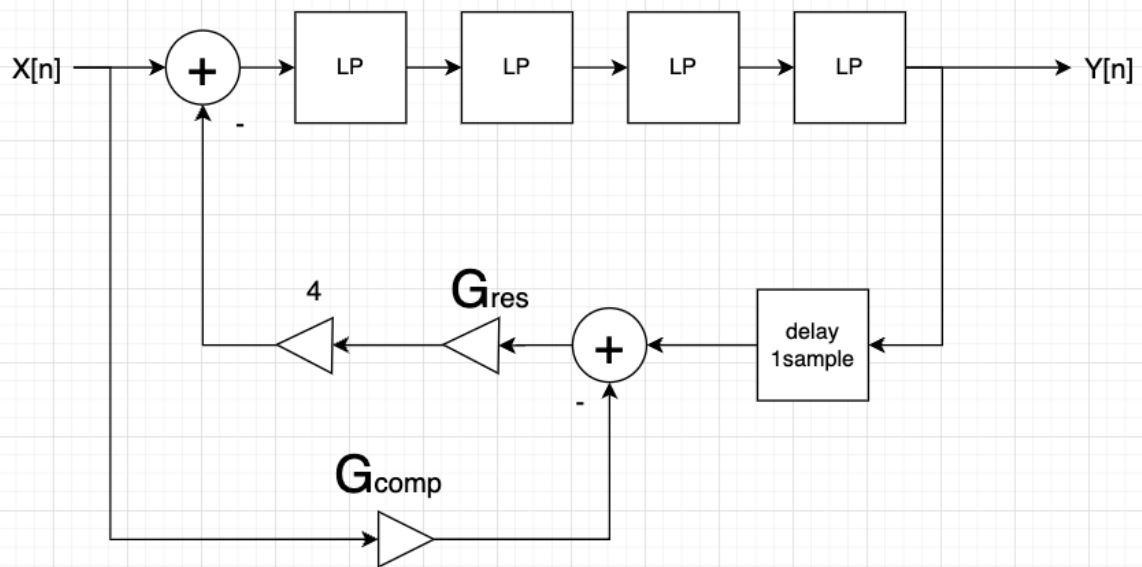
Diagrammen:

Project 1

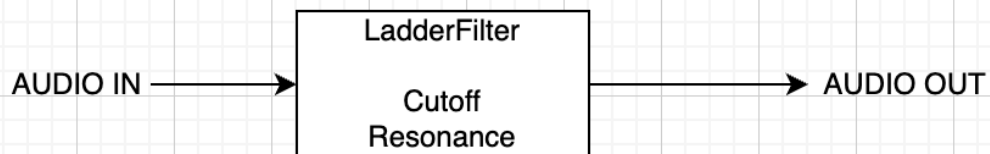


Project 2

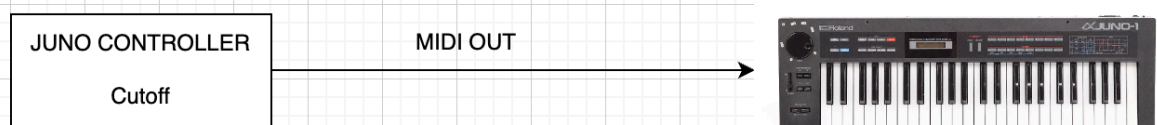




Dit is wat ik heb aan ladder filter de volledige staat in:  
[https://www.researchgate.net/publication/220386519\\_Oscillator\\_and\\_Filter\\_Algorithms\\_for\\_Virtual\\_Analog\\_Synthesis](https://www.researchgate.net/publication/220386519_Oscillator_and_Filter_Algorithms_for_Virtual_Analog_Synthesis)



Audioflow ladder filter



Midi flow controller