

# CSD2b - Design SynthSong Fabian Dekker

## Inleiding

### 1. Gekozen focus

1.1 Motivatie achter focus

### 2. Beschrijving

2.1 Motivatie achter ontwerp

2.2 MoSCoW

2.3 Beschrijving van ontwerp synth 1 en 2

### 3. User Interaction flow diagram

### 4. Pseudo code en Audio Flow FM Synth

### 5. Pseudo code en Audio Flow Additieve Synth

## **1. Gekozen focus**

Voor dit project heb ik gekozen om me te focussen op synthese. (SYNTH FOCUS)

### **1.1 Motivatie achter focus**

De motivatie achter deze keuze was dat ik dacht het meeste te kunnen leren van een focus op synthese. Bij een focus op melodie denk ik dat voor mij het project te veel zou lijken op de opdracht die we moesten doen in blok 1 (irregular beat generator). Ik wil graag toewerken naar het maken van plug-ins en het creëren van software voor muziek. Ik denk dat synthese hierbij een belangrijk onderdeel is.

## **2. Beschrijving**

### **2.1 Motivatie achter ontwerp**

Ik wil voor mijn ontwerp het KISS (keep it simple stupid) principe toepassen om te zorgen dat ik mezelf niet overwerk. Ik zal proberen tijdens het programmeren de code enigszins open te houden zodat ik gemakkelijk functies kan toevoegen. Ik heb de MoSCoW-methode gebruikt om de opdracht voor mezelf in te kaderen.

## 2.2 MoSCoW

### Must have:

- De gebruiker kan kiezen uit twee verschillende synthesizers.
  - > Additieve synth (add\_synth)
  - > FM synth (fm\_synth)
- De synthesizers hebben beide 1 parameter die door de gebruiker aangepast kan worden.
  - > Bij add\_synth is dit de hoeveelheid boventonen (1 t/m 16)
  - > Bij fm\_synth is dit de frequentie van de modulator
- De gekozen synthesizer speelt een voorprogrammeerde melodie af.

### Should have:

- De synthesizers kunnen bestuurd worden met MIDI.
- De afspeelsnelheid van de voorprogrammeerde melodie is instelbaar.
- Er zijn meerdere melodieën om uit te kiezen.
  - Twee melodieën kunnen tegelijk afspelen (e.g. bas en lead).
- Volume envelopes
  - De amplitude van de modulator (bij de fm\_synth) volgt ook de volume envelope
- De hoeveelheid boventonen die je kan instellen met de parameter van de add\_synth is veel groter.
  - > Wordt beperkt door de nyquistfrequentie.

### Could have:

- Een extra parameter bij add\_synth die de amplitude van de boventonen bepaald.
  - > Mapping is sample and hold met smoothing waarbij boventonen met een hogere frequentie steeds minder smoothing krijgen.
  - > Eventueel meerdere mapping opties
- De parameter van de gekozen synth kan bestuurd worden met OSC.
- De synth is polyfoon met 8 voices
- Het programma kan midi bestanden afspelen
- Support voor pitchbend

### Won't have:

- Grafische user interface
- Geprogrammeerde effecten als reverb, distortion etc.
- Melodie generatie
- Filters

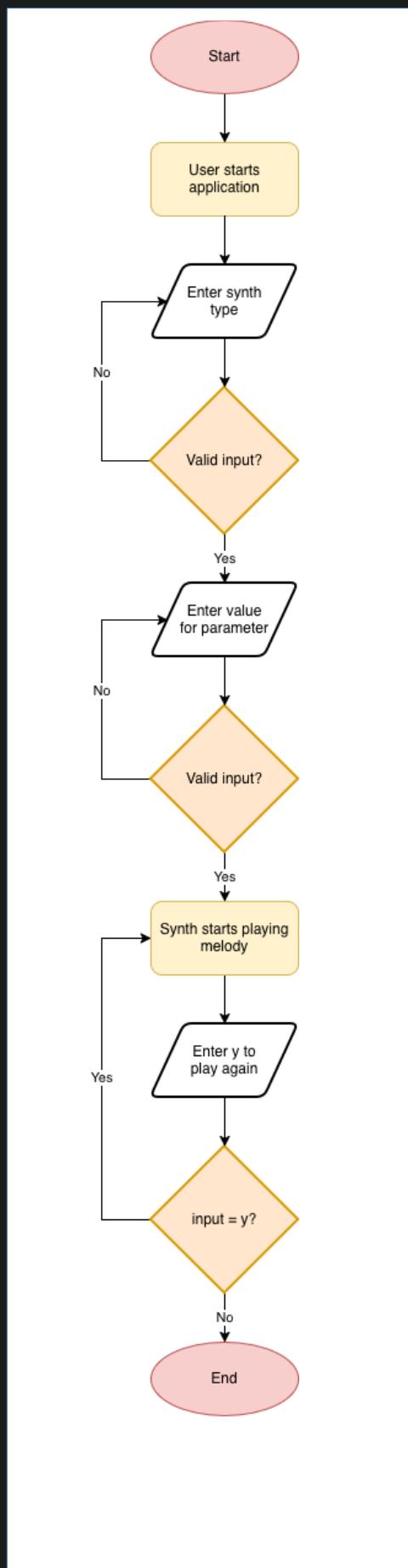
## **Beschrijving van ontwerp synth 1 en 2:**

*Disclaimer: dit is de beschrijving van mijn uitwerking van de opdracht wanneer ik me houd aan het kiss principe. De Should have's en could have's zijn hier niet in mee genomen. Ik zal mijn beschrijving updaten als het project groter wordt dan het minimum viable product.*

De user heeft keuze uit twee monofone synthesizers. Synth 1 maakt gebruik van additieve synthese. De parameter die ingesteld kan worden bij deze synthesizer is de hoeveelheid boventonen. Deze boventonen zijn in een harmonische reeks. Dus de eerste boventoon is 2x de grondtoon. De tweede boventoon is 3x de grondtoon etc. Dit kunnen maximaal 16 boventonen zijn.

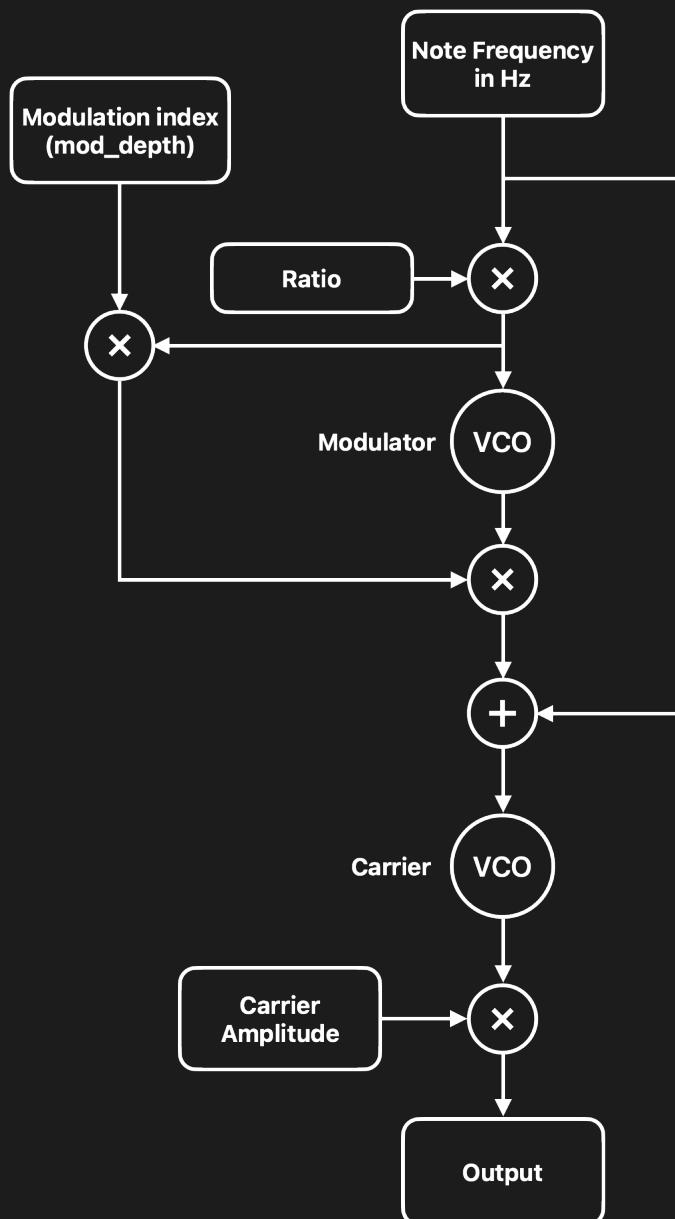
Synth 2 is een fm synthesizer met een operator. De parameter die ingesteld kan worden bij deze synthesizer is de ratio van de modulator. Beide synthesizers spelen een voorprogrammeerde melodie af.

### 3. User Interaction Flow diagram



#### 4. Pseudo code en Audio Flow FM Synth

```
// note_frequency is the frequency in Hz of note that's playing  
// mod_depth is an adaptable parameter of this synth  
  
// Initialize values  
note_frequency = 261.63f;  
ratio = 2.0f;  
mod_depth = 5.0f;  
  
// Initialize amplitudes  
carrierAmplitude = 1.0f;  
modulator_amplitude = mod_depth * (note_frequency * ratio);  
  
// Code per sample  
modulator.setFrequency(note_frequency * ratio);  
modulator_signal = modulator.getSample() * modulator_amplitude;  
carrier.setFrequency(note_frequency + modulator_signal);  
output = carrier.getSample();
```



## 5. Pseudo code en Audio Flow Additieve synth

```
//Amount of overtones is an adaptable parameter of the synth
overtones_amt = 16;
amplitude = 1/overtones_amt;
osc_array = [];

for i in overtones_amt:
    osc_array.push(oscillator.setFrequency(frequency * i));
    osc_array[i].setAmplitude(amplitude);
    output = osc_array[i].getSample();
```

