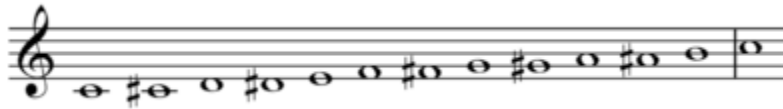


In de derde les van Unit 2 leer je de verbanden tussen de muzieknoten en ga je een programma schrijven dat deze muzieknoten laat horen.

Doelen:

- Uitleg over toonladder en de relatie met $\sqrt[12]{2}$
- Een programma maken dat een toonladder speelt.



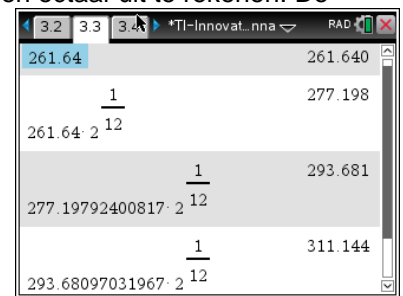
Muziektheorie

Muzieknoten worden bepaald door de frequentie van trillende objecten zoals een luidspreker, een trommelvel of een snaar (zoals van een gitaar of een piano). De noten van een (chromatische) toonladder hebben een speciaal wiskundig verband. Er zijn 12 stappen in een octaaf om van een toon in het ene octaaf naar een dezelfde toon in het volgende octaaf te gaan. Als een toon een frequentie f heeft dan heeft de volgende toon in de toonladder een frequentie van $f \times \sqrt[12]{2}$

Als je de frequentie van een toon 12 keer vermenigvuldigd met $\sqrt[12]{2}$ of $2^{(1/12)}$ (de twaalfde machtswortel van 2) dan geeft dit een verdubbeling van de originele frequentie. Dit betekent dat de 12-de noot in een toonladder een frequentie heeft van $f \times \sqrt[12]{2}^{12} = 2 \times f$. De frequentie van de lage toon is de helft van de frequentie van de toon een octaaf hoger. Bijvoorbeeld bij een toon met een frequentie van 440 Hz heeft dezelfde toon een octaaf hoger een frequentie van 880 Hz, en een octaaf lager heeft een frequentie van 220 Hz.

Het menselijk oor ervaart tonen die een octaaf uit elkaar liggen toch als “dezelfde” toon. Dit wordt veroorzaakt door de overeenkomstige hogere harmonische trillingen. Omdat het menselijk oor deze tonen als “hetzelfde” waarneemt hebben deze tonen dezelfde naam gekregen in het Westers muzieksysteem. De naam van een toon een octaaf boven een C is ook een C. De intervallen tussen de tonen van een chromatische toonladder worden halve toonsafstanden genoemd.

In dit project gebruiken we het $\sqrt[12]{2}$ principe om de frequenties van de twaalf tonen in een octaaf uit te rekenen. De C in het ééngestreept octaaf (de eerste noot in het notenvoorbeeld hierboven) heeft een frequentie 261.64 Hz. De C in het tweegestreept octaaf heeft een frequentie van 2×261.64 Hz dus 523.28 Hz. Dit is de laatste noot in het notenvoorbeeld hierboven. Er zijn 12 stappen (halve noten) tussen deze twee tonen en elke toon heeft een frequentie $2^{1/12}$ x de frequentie van de toon ervoor.



In het figuur hiernaast hebben we het getal 261.64 in getikt in de rekenmachine en in de volgende regel hebben we dit vermenigvuldigd met $2^{1/12}$.

De rekenmachine voegt **Ans** in als je begint met vermenigvuldigen omdat je altijd minimaal 2 getallen met elkaar vermenigvuldigt. Als je een aantal maal op enter klikt dan krijg je de antwoorden zoals in het figuur. Als je 12 keer doorgaat met enter drukken dan wordt het antwoord 523.28, precies 2 x de beginwaarde, omdat $f \times (2^{1/12})^{12} = 2 \times f$.

In het programma gaan we gebruik maken van deze herhalende eigenschappen van noten.

Tip: Voor leerlingen die niet bekend zijn met wortels en machten kan dit ingewikkelds zijn. Het kan handig zijn om dit principe op een piano of keyboard te demonstreren. Je kunt dan de octaven laten horen, maar ook de twaalf tonen er tussen in. Verdere informatie is ook te vinden in BiNaS tabel 15 C.

Het programma:

1. Start een nieuw programma en noem het SOUND2
2. Voeg **Disp** toe met tussen haakjes "Toonladder"
3. Geef de variabele f de waarde 261.64
4. Deze variabele zal alle frequenties krijgen van de twaalf noten in de toonladder.

```

"sound2" succesvol opgeslagen
Define sound2()=
Prgm
Disp "Toonladders"
f:=261.64
[]
EndPrgm
    
```

De For Loop:

5. Maak een **For** loop die loopt van 1 tot 13 (voor de twaalf noten in de ladder plus het octaaf)
6. Voeg het commando **Send** "SET SOUND toe uit het **HUB** menu
7. Voeg **eval()** toe met de variabele f zoals in het voorbeeld.

```

"sound2" succesvol opgeslagen
Define sound2()=
Prgm
Disp "Toonladders"
f:=261.64
For i,1,13
Send "SET SOUND eval(f)"
[]
EndFor
EndPrgm
    
```

Aanpassen van de frequentie:

8. Vermenigvuldig f steeds met $2^{1/12}$ en sla de waarde weer op in f
 $f := f * 2^{(1/12)}$
 Deze regel neemt de huidige waarde van f en vermenigvuldigd dit met $2^{1/12}$ en slaat deze nieuwe waarde weer op in f .
9. Voer het programma uit met ctrl+R

```

"sound2" succesvol opgeslagen
Define sound2()=
Prgm
Disp "Toonladders"
f:=261.64
For i,1,13
Send "SET SOUND eval(f)"
f:=21/12.f
[]
EndFor
EndPrgm
    
```

Aanpassingen aan het Programma:

Voeg een **TIME** parameter toe aan het commando **Send** "SET SOUND en een even lang **Wait** commando om er voor te zorgen dat elke noot gespeeld kan worden. Zodra de TI-Innovator Hub een nieuw commando gaat hij verder met het volgende commando. Hij maakt het eerdere niet commando eerst af.

```

"sound2" succesvol opgeslagen
Define sound2()=
Prgm
Disp "Toonladders"
f:=261.64
For i,1,13
Send "SET SOUND eval(f) TIME 0.5"
Wait 0.5
f:=21/12.f
[]
EndFor
EndPrgm
    
```