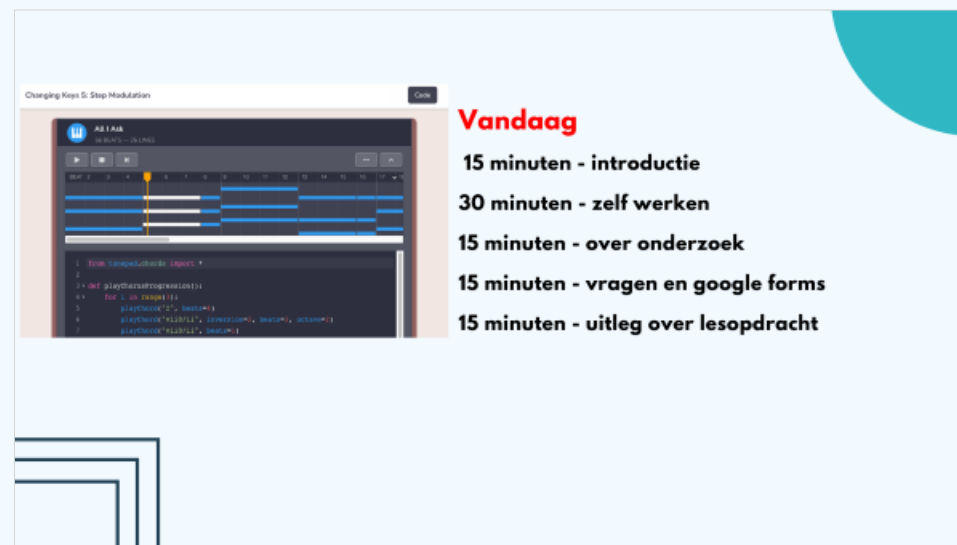


# Presentatie i&i conferentie 2022

## Python leren met muziek



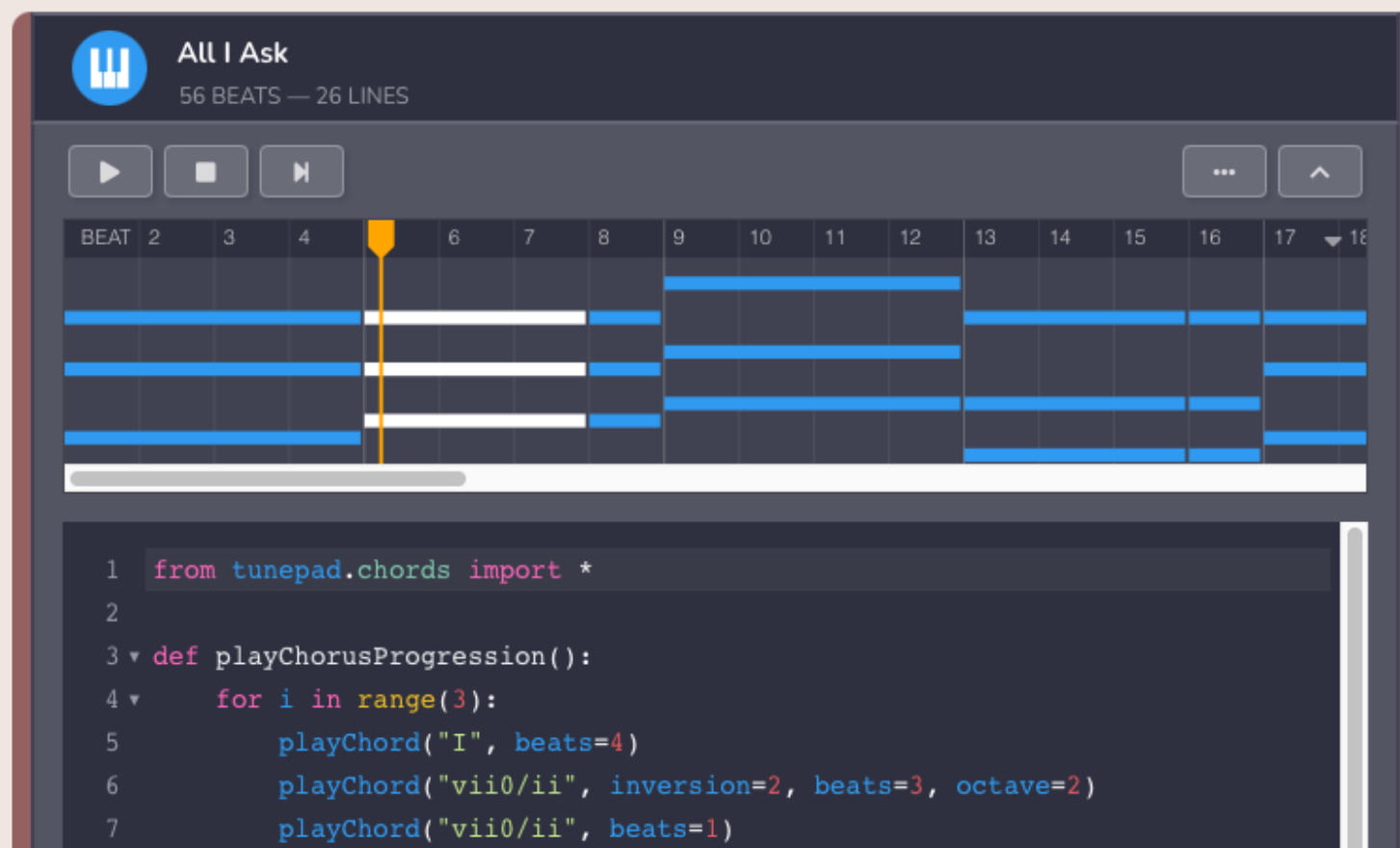
Changing Keys 5: Step Modification

```
def play_chords(chords):  
    for chord in chords:  
        play_chord(chord)
```

**Vandaag**

- 15 minuten - introductie
- 30 minuten - zelf werken
- 15 minuten - over onderzoek
- 15 minuten - vragen en google forms
- 15 minuten - uitleg over lesopdracht

Krijn Hoogendorp



The screenshot shows a music software interface. At the top, it says "All I Ask" and "56 BEATS — 26 LINES". Below this is a piano roll with a timeline from BEAT 2 to 18. A yellow vertical line is positioned at beat 5. The piano roll shows several blue horizontal bars representing notes or chords across different staves. Below the piano roll is a code editor with the following Python code:

```
1 from tunepad.chords import *
2
3 def playChorusProgression():
4     for i in range(3):
5         playChord("I", beats=4)
6         playChord("vii0/ii", inversion=2, beats=3, octave=2)
7         playChord("vii0/ii", beats=1)
```

## Vandaag

**15 minuten - introductie**

**30 minuten - zelf werken**

**15 minuten - over onderzoek**

**15 minuten - vragen en google forms**

**15 minuten - uitleg over lesopdracht**

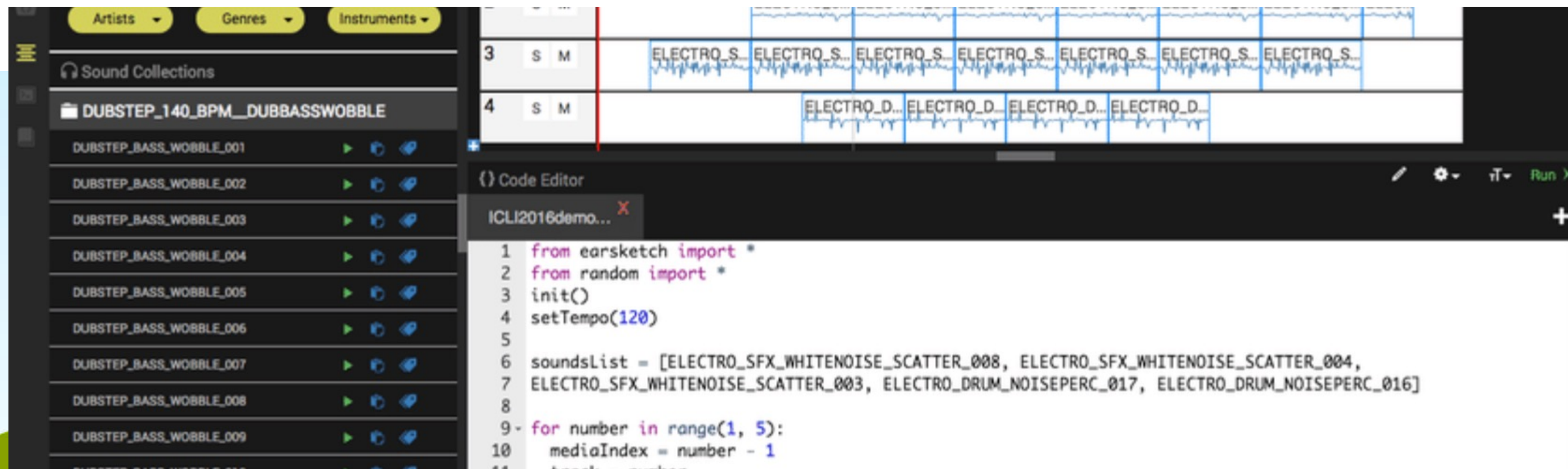
# Krijn Hoogendorp

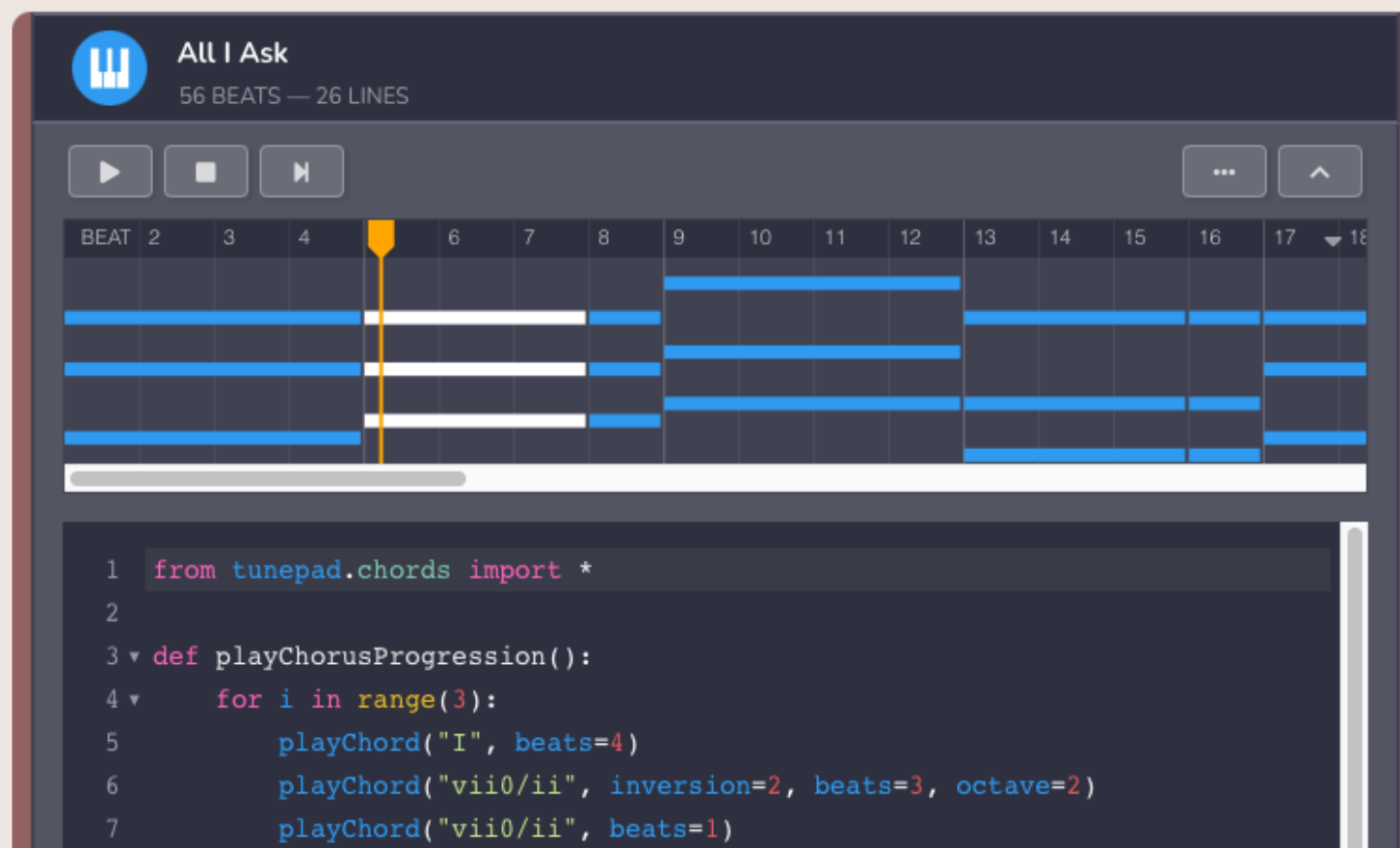
**k.hoogendorp@hva.nl**

Informaticadocent sinds 2016 (middelbare school vmbo/havo/vwo), mbo en hbo.

Hiervoor oa before specialist en project manager ABN AMRO/IBM.

Momenteel werkzaam bij Hogeschool van Amsterdam (Cyber Security) en 1 dag p/w bij ROC van Amsterdam (opleiding Drones & Engineering)





The screenshot shows a music software interface for a piece titled "All I Ask" with 56 beats and 26 lines. The piano roll displays a progression of chords across 18 beats. A yellow vertical line is positioned at beat 5. Below the piano roll, a code editor shows the following Python code:

```
1 from tunepad.chords import *
2
3 def playChorusProgression():
4     for i in range(3):
5         playChord("I", beats=4)
6         playChord("vii0/ii", inversion=2, beats=3, octave=2)
7         playChord("vii0/ii", beats=1)
```

**Is het mogelijk om programmeren te leren met muziek als context?**

```
1 from tunepad.chords import *  
2  
3 def playChorusProgression():  
4     for i in range(3):  
5         playChord("I", beats=4)  
6         playChord("vii0/ii", inversion=2, beats=3, octave=2)  
7         playChord("vii0/ii", beats=1)
```

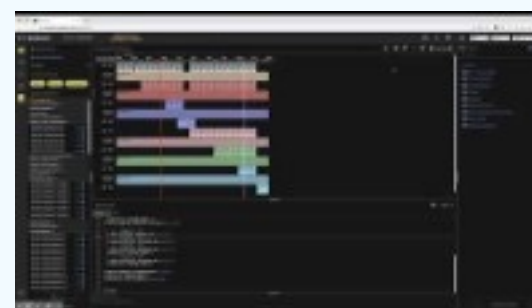
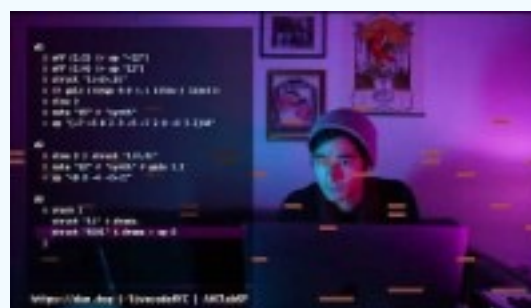
**Is het mogelijk om programmeren te leren met muziek als context.**

Ja:

[Sonic Pi \(Ruby\)](#)

[EarSketch \(Python/JavaScript\)](#)

[TunePad \(Python\)](#)



New Bass  
4 BEATS — 5 LINES

▶ ■ ▶⏸ ⋮

BEAT	2	3	4
			█
█	█	█	

```
1 # New TunePad cell
2 playNote(1)
3 playNote(1)
4 playNote(1)
5 playNote(8).
```

808 DRUMKIT ▾ REC ● MIDI ●

KICK 0	KICK 1	SNARE 2	SNARE 3	HAT 4	HAT 5
TOM 6	TOM 7	TOM 8	CYMBAL 9	CLAP 10	SHAKER 11

*playNote(1)* laat een geluid horen  
In dit geval een 'kick'

Is dat heel anders dan het printen  
van een string?

*print("hallo")*

NEW Bass  
4 BEATS — 5 LINES

BEAT 2 3 4

```
1 # New TunePad cell  
2 playNote(1)  
3 playNote(1)  
4 playNote(1)  
5 playNote(8).
```

808 DRUMKIT ▾ REC MIDI

KICK KICK SNARE SNARE HAT HAT  
0 1 2 3 4 5

TOM TOM TOM CYMBAL CLAP SHAKER  
6 7 8 9 10 11

*playNote(1)* laat een geluid horen  
In dit geval een ‘kick’

Is dat heel anders dan het printen van  
een string?

*print("hallo")*

**Voorbeeld:**

<https://tunepad.com/project/37117>

NEW Bass  
4 BEATS — 5 LINES

▶ ■ ▶⏸ ⋮

BEAT	2	3	4

```
1 # New TunePad cell
2 playNote(1)
3 playNote(1)
4 playNote(1)
5 playNote(8).
```

808 DRUMKIT ▾ REC MIDI

KICK 0	KICK 1	SNARE 2	SNARE 3	HAT 4	HAT 5
TOM 6	TOM 7	TOM 8	CYMBAL 9	CLAP 10	SHAKER 11

*Heel veel constructies die in programmeren gangbaar zijn, kun je ook met muziek uitleggen.*



The screenshot shows a music programming interface. At the top, it says "NEW Bass" and "4 BEATS — 5 LINES". Below this are playback controls: a play button, a stop button, a next button, and a menu button. The main area is a 4-beat sequence grid. The first three beats have a green bar at the bottom, and the fourth beat has a green bar at the top. Below the grid is a code editor with the following code:

```
1 # New TunePad cell
2 playNote(1)
3 playNote(1)
4 playNote(1)
5 playNote(8).
```

At the bottom, there is a drum kit interface with a dropdown menu set to "808 DRUMKIT", a "REC" button, and a "MIDI" button. The drum kit consists of 12 buttons labeled 0 through 11, with the following names:

0	KICK	1	KICK	2	SNARE	3	SNARE	4	HAT	5	HAT
6	TOM	7	TOM	8	TOM	9	CYMBAL	10	CLAP	11	SHAKER

*Heel veel constructies die in programmeren gangbaar zijn, kun je ook met muziek uitleggen.*

*List:*

*Een C schaal = [60, 62, 64, 65, 67, 69, 71, 72]*

NEW Bass  
4 BEATS — 5 LINES

▶ ■ ▶⏪ ⋮

BEAT	2	3	4

```
1 # New TunePad cell
2 playNote(1)
3 playNote(1)
4 playNote(1)
5 playNote(8)
```

808 DRUMKIT ▾ REC MIDI

KICK 0	KICK 1	SNARE 2	SNARE 3	HAT 4	HAT 5
TOM 6	TOM 7	TOM 8	CYMBAL 9	CLAP 10	SHAKER 11

***Heel veel constructies die in programmeren gangbaar zijn, kun je ook met muziek uitleggen.***

***Een list met een for-loop***

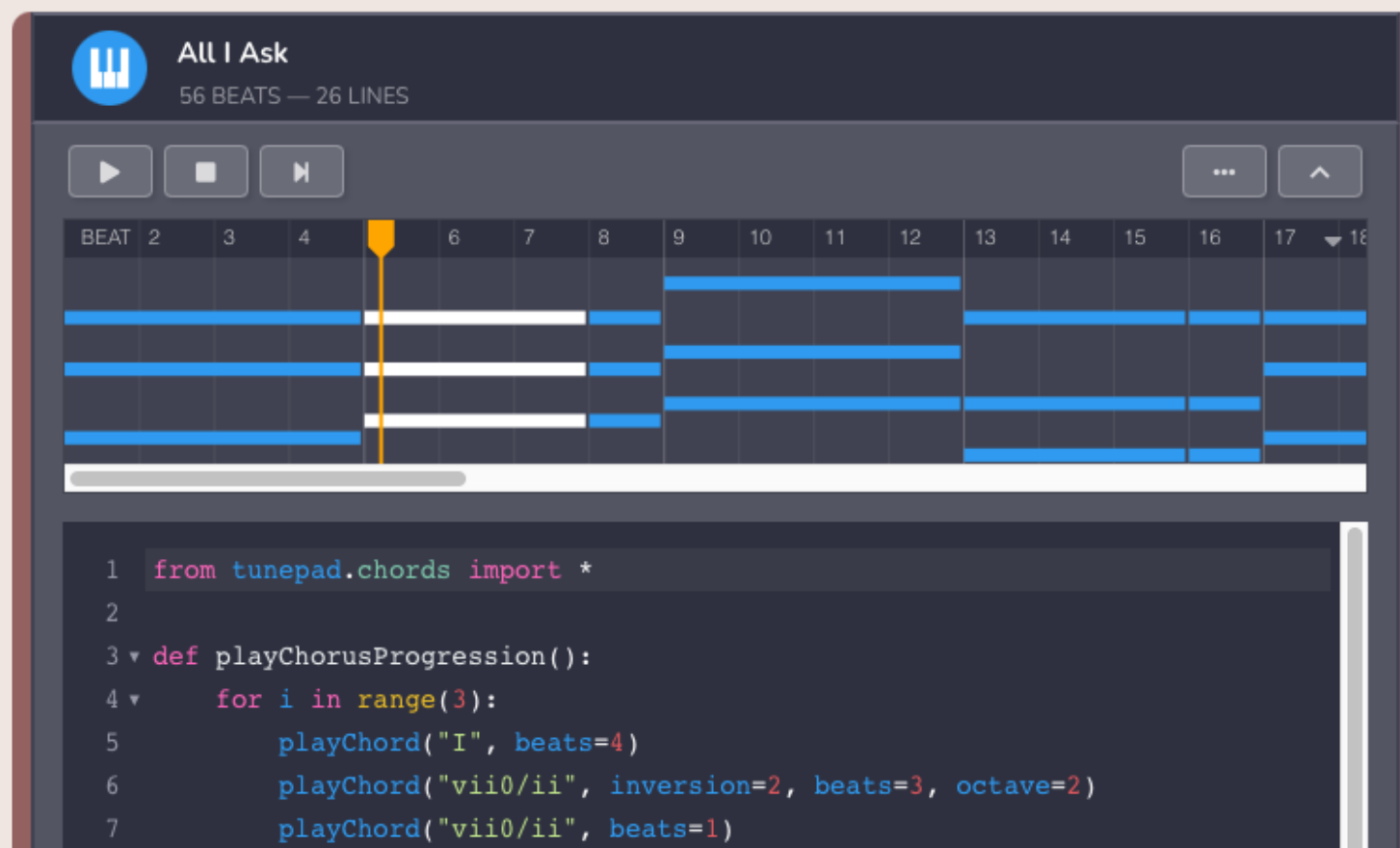
***C\_schaal = [60, 62, 64, 65, 67, 69, 71, 72]***

***For i in C\_schaal:***

***playNote(i)***

***Niet alleen lists en variabelen, maar ook functies***

***Voorbeeld van student: [New Jam Session \(tunepad.com\)](https://tunepad.com)***

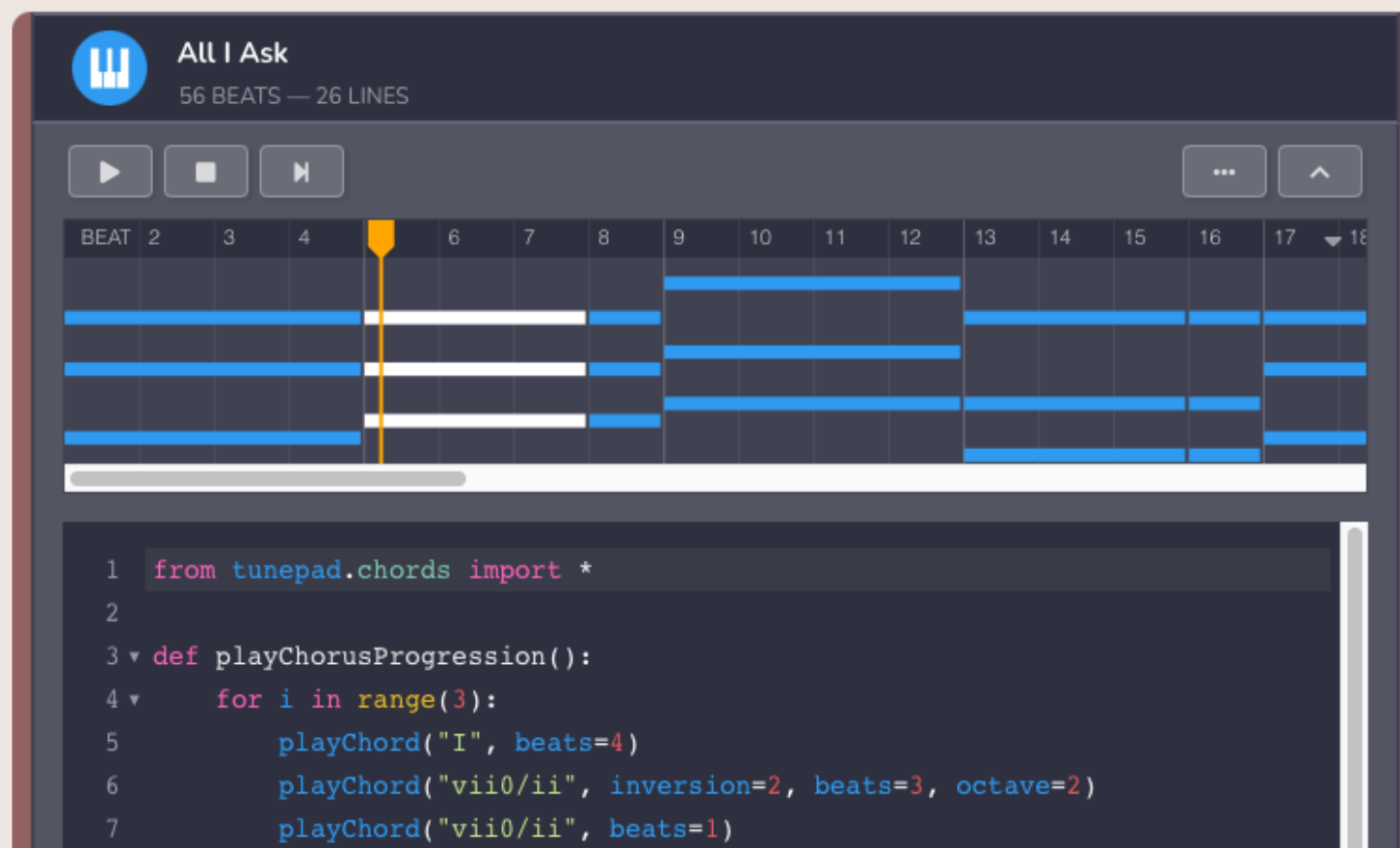


The screenshot shows a music software interface for a piece titled "All I Ask" with 56 beats and 26 lines. The piano roll displays a progression of chords across 18 beats. A yellow cursor is positioned at beat 5. Below the piano roll, a code editor shows the following Python code:

```
1 from tunepad.chords import *
2
3 def playChorusProgression():
4     for i in range(3):
5         playChord("I", beats=4)
6         playChord("vii0/ii", inversion=2, beats=3, octave=2)
7         playChord("vii0/ii", beats=1)
```

## Maar waarom muziek als context gebruiken?

- Time on task
- Low floor/high ceiling/wide walls
- Geschikt voor verschillende leeftijden/niveaus



The screenshot shows a music software interface for a piece titled "All I Ask" with 56 beats and 26 lines. The piano roll displays a progression of chords across 18 beats. A yellow vertical line is positioned at beat 5. Below the piano roll, a code editor shows the following Python code:

```
1 from tunepad.chords import *
2
3 def playChorusProgression():
4     for i in range(3):
5         playChord("I", beats=4)
6         playChord("vii0/ii", inversion=2, beats=3, octave=2)
7         playChord("vii0/ii", beats=1)
```

## Maar waarom muziek als context gebruiken?

- Time on task
- Low floor/high ceiling/wide walls
- Geschikt voor verschillende leeftijden/niveaus
- MAAR: ook nadelen(?)

## Probleem met populatie van Informatica

- Meer mannen dan vrouwen
- 'Ethnically skewed' (GeorgiaTech onderzoeken)
- 'computer science meer geschikt voor 'beta' ' leerlingen.
- Informatie is vaak alleen mogelijk voor NT en NG profielen.

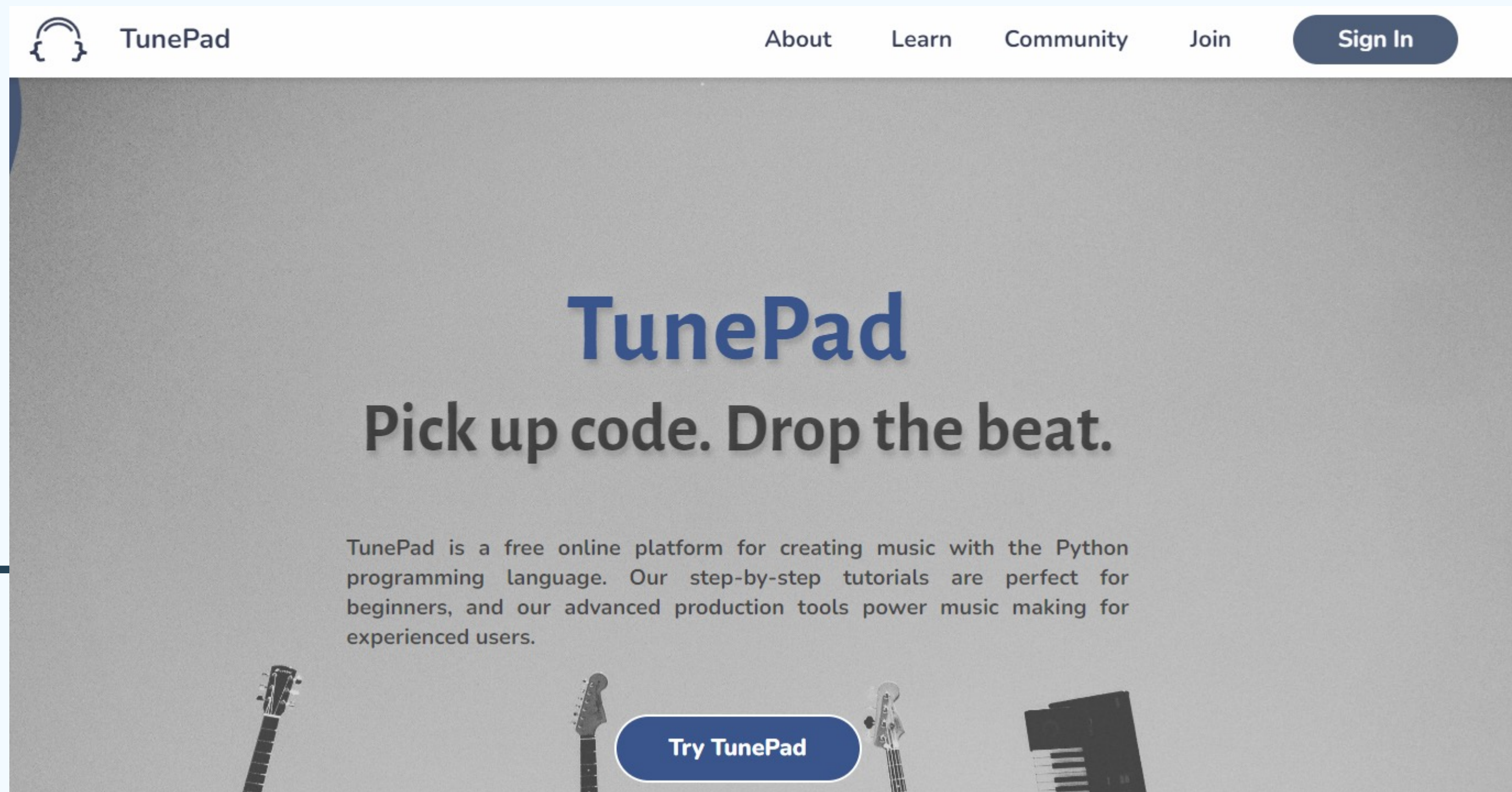
## Waarom is populatie niet divers?

Misschien vanwege imago en geschiedenis van van het vakgebied.  
Veel opdrachten zijn gerelateerd aan wiskundige problemen zoals  
'kortste pad' of 'toren van Hanoi'.


Uitdagend voor sommigen  
Niet uitdagend voor velen.

# Zelf aan de slag

<https://tunepad.com/>



The screenshot shows the homepage of the TunePad website. At the top left is the TunePad logo, which consists of a pair of headphones with curly braces on either side, followed by the text "TunePad". To the right of the logo are navigation links: "About", "Learn", "Community", and "Join". A dark blue button with the text "Sign In" is positioned to the right of these links. The main content area has a dark grey background. In the center, the word "TunePad" is written in a large, bold, blue font. Below it, the tagline "Pick up code. Drop the beat." is written in a bold, dark grey font. Underneath the tagline is a paragraph of text: "TunePad is a free online platform for creating music with the Python programming language. Our step-by-step tutorials are perfect for beginners, and our advanced production tools power music making for experienced users." At the bottom of the page, there are images of a guitar, a bass, and a keyboard. A dark blue button with the text "Try TunePad" is centered at the bottom, overlapping the images.

 TunePad

[About](#) [Learn](#) [Community](#) [Join](#) [Sign In](#)

# TunePad

## Pick up code. Drop the beat.

TunePad is a free online platform for creating music with the Python programming language. Our step-by-step tutorials are perfect for beginners, and our advanced production tools power music making for experienced users.

[Try TunePad](#)

# Research

**The effectiveness and attractiveness  
of music as a context to learn to code**





## Importance of programming

Computer knowledge is form of literacy, when societies become dependent on that knowledge, those who can code have power and influence on those who cannot. (based on Horn et al., 2020)

What are schools for?

*“... powerful disciplinary knowledge .”*

*“...what the knowledge can do or what intellectual power it gives to those who have access to it.”*

(Michael Young, 2008)

# Importance of diverse student population

But in professional field:

- not so much math/statics,
- also design,
- and programming is currently present in most domains ('omnipresent')

Development of e.g. *arts* as a context to teach coding



## But is context *arts*,

- attractive enough to diversify the student population?,
- 'computational' enough?, and
- will concepts learned be transferable to other contexts?

## Is the context music suitable for teaching coding.

- Will it produce sufficient content-knowledge on coding?
- are this skills and knowledge transferable from one domain (music) to other domains?
- What aspects music can ensure a permanent and flexible knowledge of coding concepts?

The research will be about the effectiveness and attractiveness of music as a context to learn how to code.

## Themes to be researched:

- a. Is music an **appropriate context** for teaching specific programming concepts and aptitude (and are there certain characteristics in music that hamper learning how to code).
- b. How will the characteristics and complexity of music influence the **process** of learning.
- c. What are requirements to come to efficient **transfer** of knowledge between different programming domains.
- d. What are the aspects of music that can promote **computational thinking**.

## a. Is music an appropriate context

What characteristics in music that hamper or help learning how to code.

An efficient context should (Nijenhuis-Voogt et al, 2020):

- be meaningful,
- in line with student interests
- usefull for learning objectives.

## Is music a meaningful context

Meaningful can have characteristics:

- part of daily experience,
- should make sense,
- students can be actively involved.

**Is music in line with student interests.**

Yes, but .....





## Is useful for learning objectives

According to Bell and Bell (2018) “...there are related forms of thinking...”

## b. Characteristics and complexity of music as a context

- The complexity of a context is of importance as the time needed for understanding the context can not be used for the actual learning objectives (Guzdial, 2010).
- Cognitive load: intrinsic (complexity of information) extraneous (complexity of instruction) en germane (learner characteristics) (in Nijenhuis, 2020).

**What is the effect on learning, when students do not only get information on the programming concepts but also need to learn the basics of music.**

Part of this question is the role of the teachers. Can they work with such a context. What are the boundaries.

## b. Characteristics and complexity of music as a context

- The complexity of a context is of importance as the time needed for understanding the context can not be used for the actual learning objectives (Guzdial, 2010).
- Cognitive load: intrinsic (complexity of information) extraneous (complexity of instruction) en germane (learner characteristics) (in Nijenhuis, 2020).

**What is the effect on learning, when students do not only get information on the programming concepts but also need to learn the basics of music.**

Part of this question is the role of the teachers. Can they work with such a context. What are the boundaries.

*“EarSketch learners are able to compose music and learn how to program without the added barrier of entry of learning music theory about harmony, melody, chord progressions” (Magerko et al., 2016).*

## c. Requirements of efficient transfer of knowledge.

Transfer of knowledge between domains will not happen naturally.

- Organized and coherent understanding of the material is needed
- In instructions explicit attention to the underlying structure of knowledge
- attention for the use in other domains.

(Bransford et al, 2020)

### **How can effective transfer of knowledge and skills happen from a music context to other contexts.**

Transfer of knowledge and skills is likely to happen when:

through extensive exercise (the **low road**),

Making transfer explicit through abstraction of concepts and combine with other contexts (the **high road**).

## d. Relationship to Computational Thinking

Computational thinking (CT) is a set of problem-solving methods related tied to the field of Computer Science (Wing, 2006)

Characteristics: decomposition, pattern recognition, data representation, generalization/abstraction, and algorithms.

Bell & Bell (2018): There are common elements between music and CT.:

- notations in formal languages (music notation compared to programming languages and protocols);
- sequence (the order in which notes appear in time; and the order of notes and order of statements in a computer program)
- repetition (in music this includes repeats and computing loops)

## Results

- Results about the requirements for effective transfer of knowledge between contexts.
  - Results about the cognitive processes involved in learning how to code.
  - Research based arguments for the diversification of contexts in computer science → suggesting new opportunities for learning
- 
- Secondary result: increase attractiveness of programming

## Methods

- a *design study*, a series of lessons will be developed in which programming concepts are taught with the use of TunePad development environments.
- Partly of *interpretivist* nature, through discussions with teachers and students about the effects of learning to code through music, but also through observing students and teachers while they are working with the material.
- Partly empirical: artefacts created in the lesson series will be analyzed

Needed: a variety of secondary schools willing to experiment with the material

The results will be analyzed and compared with existing literature on learning how to code and the cognitive processes related to learning how to code and computational thinking.

## Other options?

- Will students also use their coding skills and knowledge in other contexts (other subjects). What is needed to make such transfer a success ?



## Other options?

- Will students also use their coding skills and knowledge in other contexts (e.g. other subjects). What is needed to make such transfer a success ?

**QUESTIONS/REMARKS?**

The screenshot shows the EarSketch web interface. The browser address bar displays `https://ears sketch.gatech.edu/ears sketch2/`. The interface is split into two main sections:

- Left Panel (Content Manager):** Features a search bar, filter options for Artists, Genres, and Instruments, and a list of sound collections. The 'SOUND COLLECTION (4097)' is expanded, showing items like 'CIARA\_MELANIN\_BEAT', 'CIARA\_MELANIN\_VOCALS', and 'FEATURED ARTISTS (172)'. Below this, 'ALICIA\_KEYS\_UNDERDOG\_BASS' and 'ALICIA\_KEYS\_UNDERDOG\_BODY' are also visible.
- Right Panel (DAW and Code Editor):** The top section is the DAW (Digital Audio Workstation) with playback controls (stop, play, refresh) and an 'EFFECTS' section. The bottom section is the 'CODE EDITOR' with 'BLOCKS MODE' disabled. It shows three open files: 'Loopingmyb...', 'theanalyzeT...', and 'ABAForm.py'. The active file 'Loopingmyb...' contains the following Python code:

```
1 from earsketch import *
2
3 # Setup:
4 init()
5 setTempo(120)
6
```

# Programmeren van muziek

Context

Transfer

Algorithmic thinking

◀ List

LEARNING PYTHON  
THROUGH CODING MUSIC

## LEARNING PYTHON THROUGH CODING MUSIC

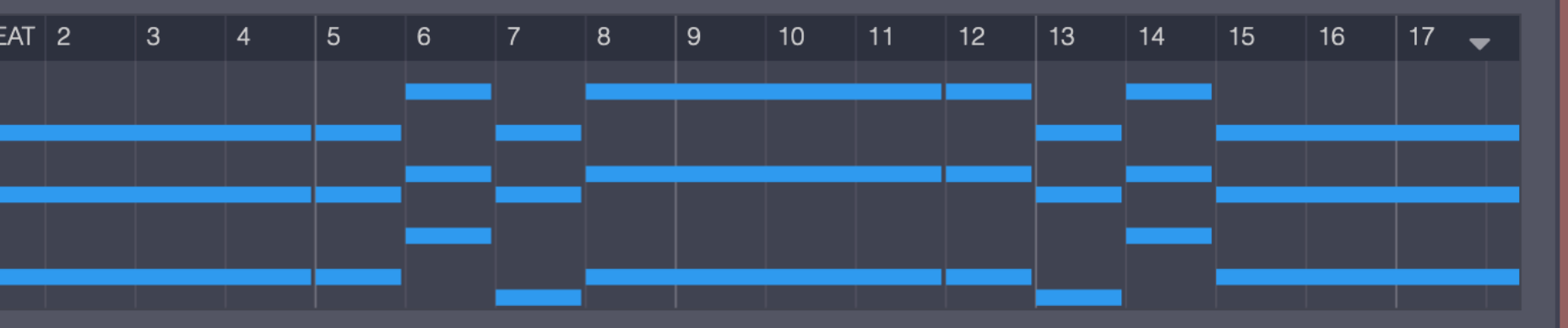
<b>Duration:</b>	14 day(s)
<b>Target group:</b>	bachelors (yr2)
<b>Start:</b>	4 July 2022
<b>Fee:</b>	€ 875.00
<b>Day:</b>	Monday / Tuesday / Wednesday / Thursday / Friday
<b>Location:</b>	<u>Wibaut building (WBH)</u>
<b>Certification:</b>	Certificate

APPLY NOW 



Literature used: Introduction to Digital Music with Python Programming (2022, Horn, West & Roberts)

Piano Chords  
56 BEATS — 43 LINES



```
1 # New TunePad cell
2 Cmaj = [ 48, 52, 55 ]
3 Dmin = [ 50, 53, 57 ]
4 Emin = [ 52, 55, 59 ]
5 Fmaj = [ 53, 57, 60 ]
6 Bdim = [ 47, 50, 53 ]
7
8 playNote(Cmaj, beats = 4)
9 playNote(Cmaj)
10 playNote(Dmin)
11 playNote(Emin)
12 playNote(Fmaj, beats = 4)
13
14 playNote(Fmaj)
15 playNote(Emin)
16 playNote(Dmin)
17 playNote(Cmaj, beats = 4)
18
```

# Tunepad

Beats:

<https://tunepad.com/featured/5>

Chords/lists:

<https://tunepad.com/project/29648>

Melody/lists

<https://tunepad.com/project/29648>



**Invullen google forms indien meer info  
gewenst (en voor inleveren product).**

**<https://forms.gle/SN3Tizkv9EBrFkpg6>**

# Opdracht

- Maak een eigen uitvoering 12 maten van 4 beats
- Schaal is [55,57,58,60,61,63,65] → G, A, Bb, C, Db, Eb, F
- Maak drum met nadruk op beat 2 en 3
- Maak basslijn met gebruik van de genoemde noten
- Maak melody met gebruik van de genoemde noten
- Het moeten 12 maten zijn van 4 beats (loops zijn mogelijk)

## Voorbeeld

<https://tunepad.com/project/37528>

<https://tunepad.com/project/37531>

## Opdracht Swing

1930s, uptempo, 4 kwartsmaat met nadruk op maat 2 en 4.

Over swing: <https://youtu.be/bGiPJZ-wRb4>

Wynton Marsalis: [https://youtu.be/\\_mLvvtV2GrA](https://youtu.be/_mLvvtV2GrA)

Zuco103 video: <https://youtu.be/jmoNxcygEdE>

## Resultaten

<https://tunepad.com/project/37528>

<https://tunepad.com/project/37531>