

Creative Systems Design

CSD2a

Vandaag

- Introductie
- Aan de slag met Git
- Aan de slag met Python

Vandaag

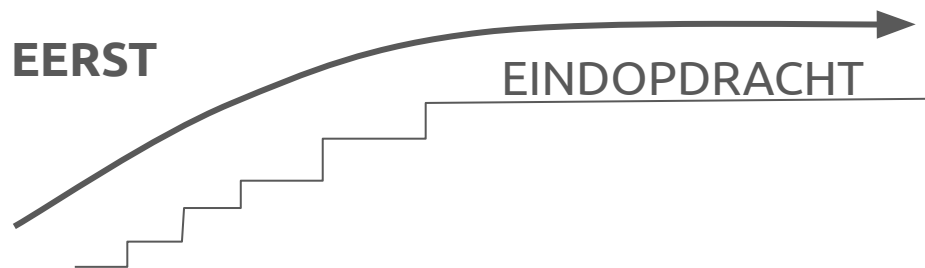
- Introductie
- Aan de slag met Git
- Aan de slag met Python

Bear with me ...

1	Introductie CSD Python Hello World en GIT intro
2	Werken met modules, basis van lists en loops
3	Lists, loops, ritmische patronen afspelen
4	Dictionary, event objecten, functions
5	Randomness en generatie van ritmische patronen
6	Ritme generatie deel 2
7	Eindopdracht voortgang, eventueel: verdiepingsonderwerpen
8	Eindpresentatie

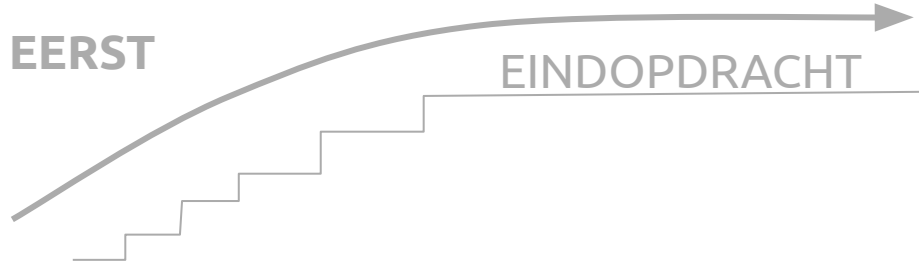
Opbouw op de schop

Omgekeerde opzet



Bottom up

Andere opzet



Bottom up



Top down & bottom up

Over CSD

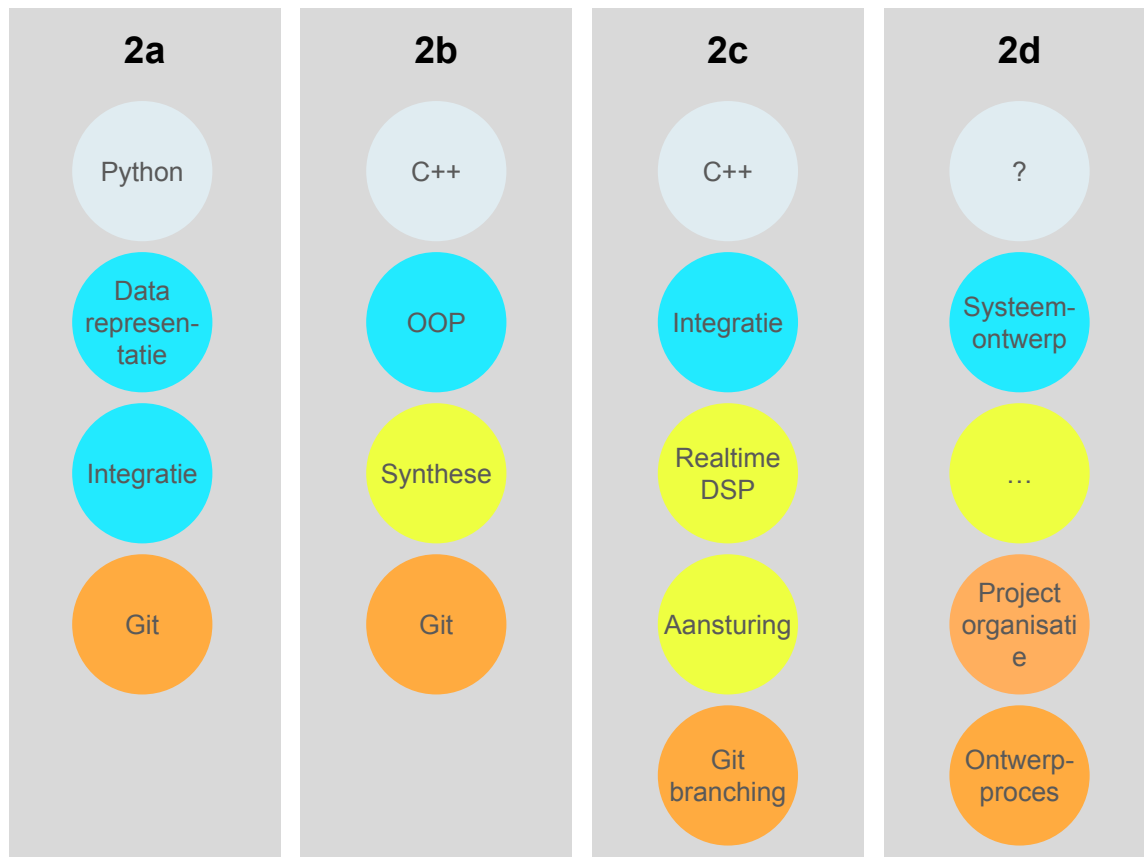
- Creative coding in verschillende MT contexten
- Techniek is dienend (*maar je moet je technieken natuurlijk wel beheersen*)
- Systeemontwerp is afhankelijk van het doel, context, beschikbare tijd etc.
- Technische ontwikkelingen → blijven ontwikkelen

CSD jaar 2

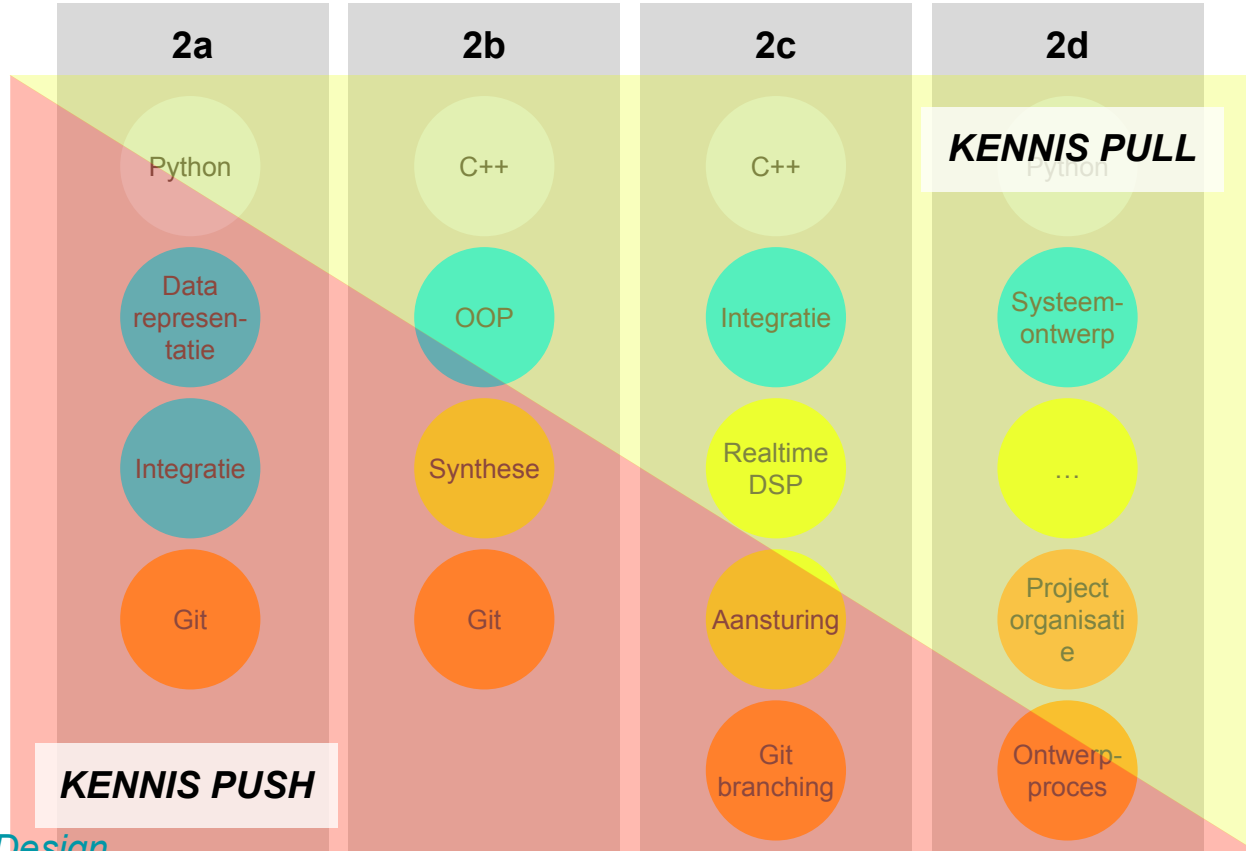
Een aantal kernwoorden:

- Artisticiteit
- Systeemontwerp, programmeertalen en hardware
- Ontwerpproces
- Verslaglegging / documentatie
- Reflectie
- Verantwoording ontwerpkeuzes

Opzet Blok 2a - CSD jaar 2



Opzet Blok 2c - CSD jaar 2



Zie ook:
Creative Systems Design

CSD 2a

1	Introductie CSD Python Hello World en GIT intro
2	Werken met modules, basis van lists en loops
3	Lists, loops, ritmische patronen afspelen
4	Dictionary, event objecten, functions
5	Randomness en generatie van ritmische patronen
6	Ritme generatie deel 2
7	Eindopdracht voortgang, eventueel: verdiepingsonderwerpen
8	Eindpresentatie

Opbouw op de schop

CSD 2a - Inzet/investering

- 6 contacturen per week
 - [Systemen bouwen met Python](#) én practicum -> veel ontwerpen en programmeren
 - [CSD Theorie](#) -> de basis
 - [Wiskunde](#) -> de basis van de basis
- 11.5 uren zelfstudie

(5 ECTS voor 8 weken → 17.5 uur per week)

Locatie content

https://github.com/ciskavriezenga/CSD_25-26 in de folder 'csd2a' :

- in **opdrachten.md** - per week de opdrachten op een rij
- in **slides/** de tijdens sessies getoonde slides
- in **handouts/** pdfs waaraan gerefereerd wordt

Locatie content

https://github.com/ciskavriezenga/CSD_25-26 in de folder 'csd2a' :

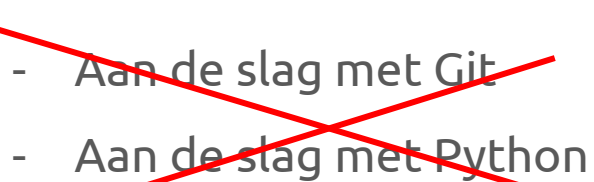
- in **opdrachten.md** - per week de opdrachten op een rij
- in **slides/** de tijdens sessies getoonde slides
- in **handouts/** pdfs waaraan gerefereerd wordt

en gezien AI met één prompt de eindopdracht genereert

→ alle code al *'gratis en voor niets'*:

- in **24-25/** de hele opzet inclusief code voorbeelden van vorig jaar

Vandaag

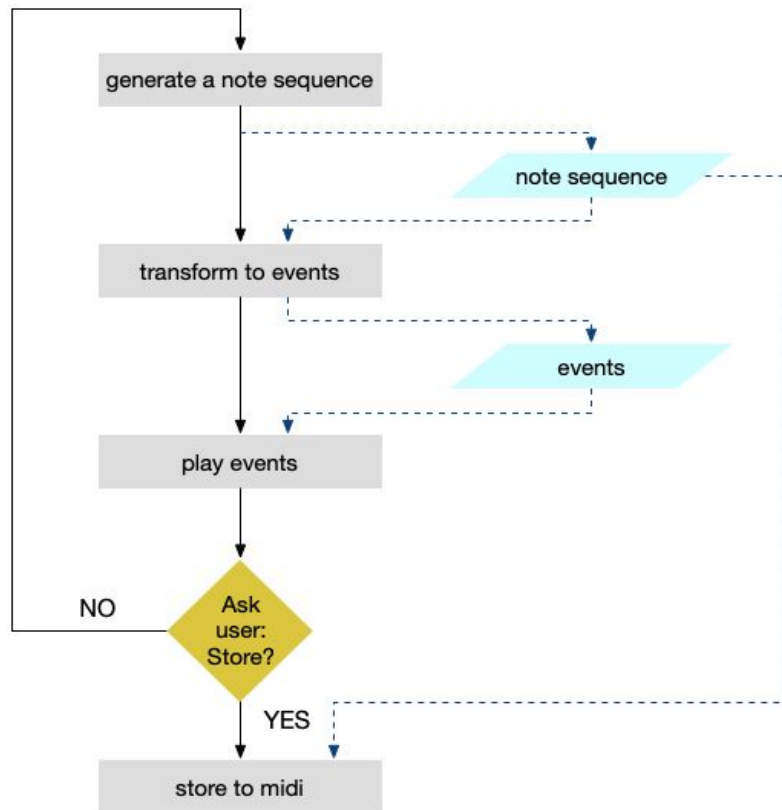
- Introductie
 - Aan de slag met Git
 - Aan de slag met Python
- 

Vandaag

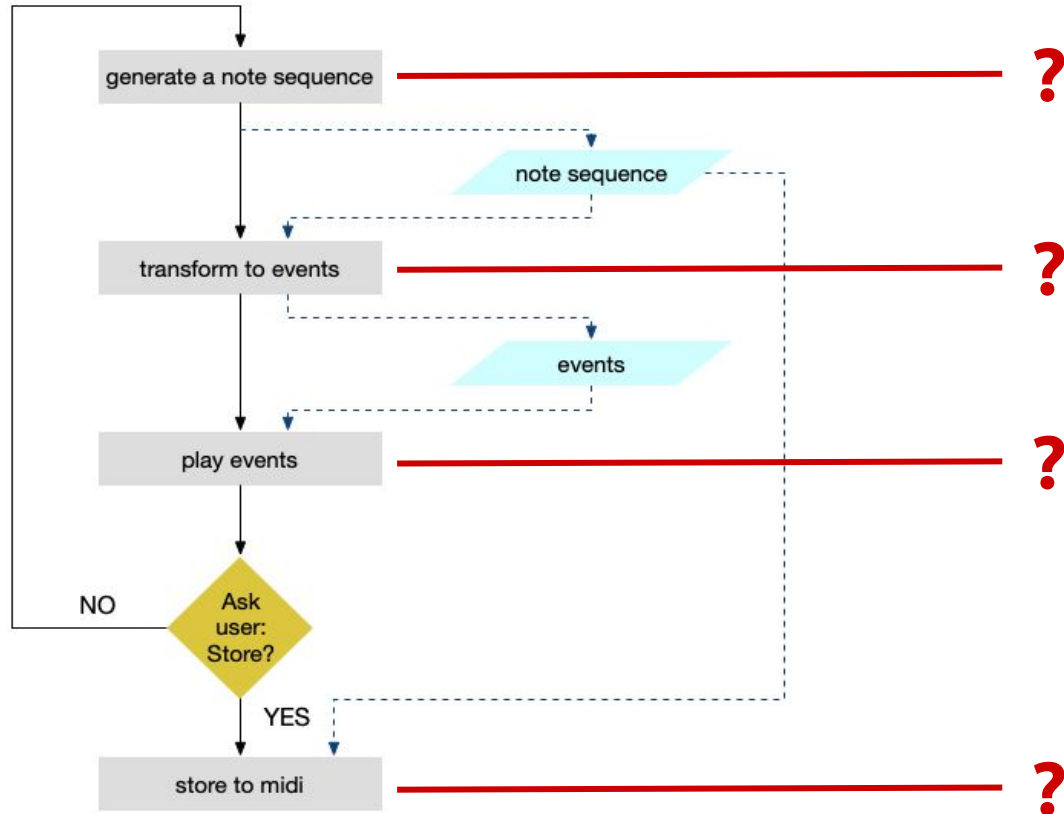
- Introductie
- Aan de slag met Git
- Aan de slag met Python
- Start met eindopdracht

ZELFSTUDIE

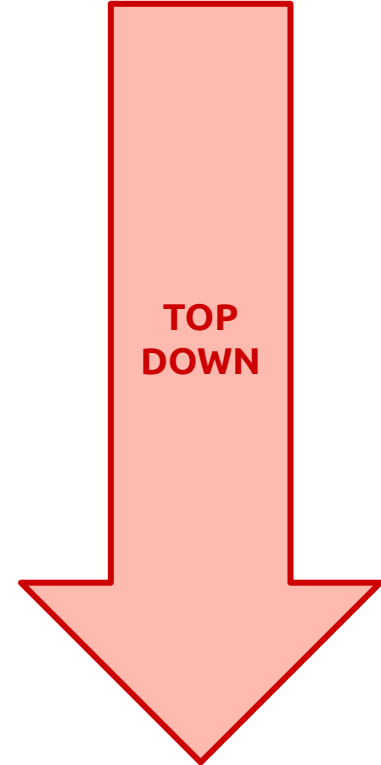
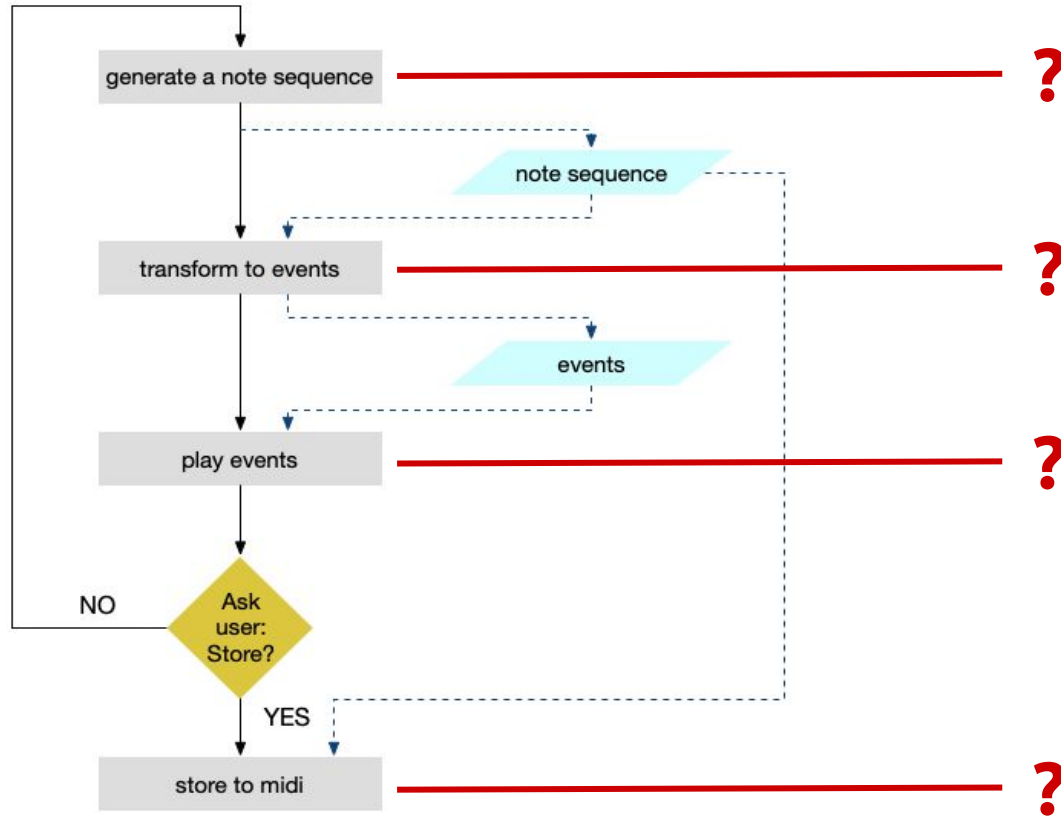
Onregelmatige maatsoort generator



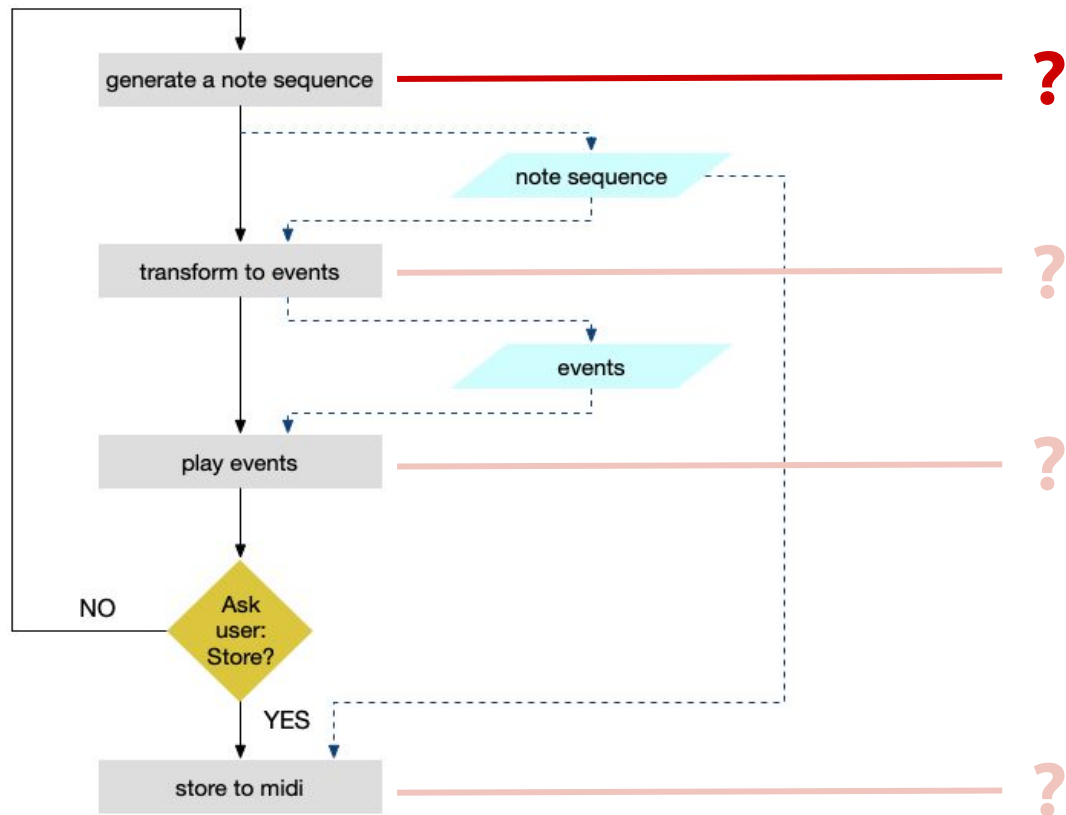
Onregelmatige maatsoort generator



Onregelmatige maatsoort generator



Onregelmatige maatsoort generator



HOE DAN?

Tot slot - *“High stakes kill creativity!”*

1	di.02-09	Introduction & example of rhythm generation strategy	Github repo, git & python up and running, DIY: design a rhythm generation strategy
2	ma.08-09	Presentations rhythm generation strategies & feedback + next steps / directions	Finalize rhythm generation strategy pseudo code and playtest it (see more elaborate description in opdrachten.md), python: bleep in a loop & create and play a mock rhythm
3	ma.15-09	Hand-in final design & playback with durations vs timestamps	implement your rhythm generation strategy to generate a note sequence (one 'stem' only for now) and then transform rhythm to timestamps
Voorheen - beoordeling op basis van werk tijdens laatste sessie			
4	ma.22-09	allow to generate and playback multiple stems by using dictionary to store events	transform your one-stem rhythm generation to multiple stems rhythm generation
5	ma.29-09	UI & trouble shooting - recap up to now	add UI to your basic irregular beat generator (IBG)
6	ma.06-10	Fancy extra stuff :-D	Beautify your IBG
7	ma.13-10	Present project up to now & presentation expectations & playtest session	Beautify your IBG to the max!
8	ma.20-10	Presentations main assignment	Practicum after final presentation? --> no practicum in that case

Tot slot - *“High stakes kill creativity!”*

→ **beoordeling**

1	di.02-09	Introduction & example of rhythm generation strategy	Github repo, git & python up and running, DIY: design a rhythm generation strategy
2	ma.08-09	Presentations rhythm generation strategies & feedback + next steps / directions	Finalize rhythm generation strategy pseudo code and playtest it (see more elaborate description in opdracht.md), python: bleep in a loop & create and play a mock rhythm
3	ma.15-09	Hand-in final design & playback with durations vs timestamps	implement your rhythm generation strategy to generate a note sequence (one 'stem' only for now) and then transform rhythm to timestamps
4	ma.22-09	allow to generate and playback multiple stems by using dictionary to store events	transform your one-stem rhythm generation to multiple stems rhythm generation
5	ma.29-09	UI & trouble shooting - recap up to now	add UI to your basic irregular beat generator (IBG)
6	ma.06-10	Fancy extra stuff :-D	Beautify your IBG
7	ma.13-10	Present project up to now & presentation expectations & playtest session	Beautify your IBG to the max!
8	ma.20-10	Presentations main assignment	Practicum after final presentation? --> no practicum in that case

Tot slot - *“High stakes kill creativity!”*

→beoordeling

1	di.02-09	Introduction & example of rhythm generation strategy	Github repo, git & python up and running, DIY: design a rhythm generation strategy
2	ma.08-09	Presentations rhythm generation strategies & feedback + next steps / directions	Finalize rhythm generation strategy pseudo code and playtest it (see more elaborate description in opdrachten.md), python: bleep in a loop & create and play a mock rhythm
3	ma.15-09	Hand-in final design & playback with durations vs timestamps Maar ... iets dat niet werkt kan niet beoordeeld worden.	implement your rhythm generation strategy to generate a note sequence (one 'stem' only for now) and then transform rhythm to timestamps
4	ma.22-09	allow to generate and playback multiple stems by using dictionary to store events if(not_working_in_7 && not_working_in_8):	transform your one-stem rhythm generation to multiple stems rhythm generation
5	ma.29-09	UI & trouble shooting setup to v result = NA	add UI to your basic irregular beat generator (IBG)
6	ma.06-10	Fancy extra stuff :-D	Beautify your IBG
7	ma.13-10	Present project up to now & presentation expectations & playtest session	Beautify your IBG to the max!
8	ma.20-10	Presentations main assignment	Practicum after final presentation? --> no practicum in that case