

Creative Systems Design

CSD2b

Vandaag

- Introductie
 - Eindopdracht: SynthSong
 - Verwachte tijdbesteding blok 2b
 - Opzet blok 2b
 - Content en bronnen
 - Beoordeling; design, systeem ontwerp, leerdoelen en reflectie en codebase
- Aan de slag
 - classes in C++
 - OOP
 - Association (*'has a' relationship*)
 - Inheritance (*'is a' relationship*)

Eindopdracht

Zie https://github.com/ciskavriezenga/CSD_25-26/blob/main/csd2b/Eindopdracht_SynthSong.pdf

CSD 2b - Inzet/investering per week

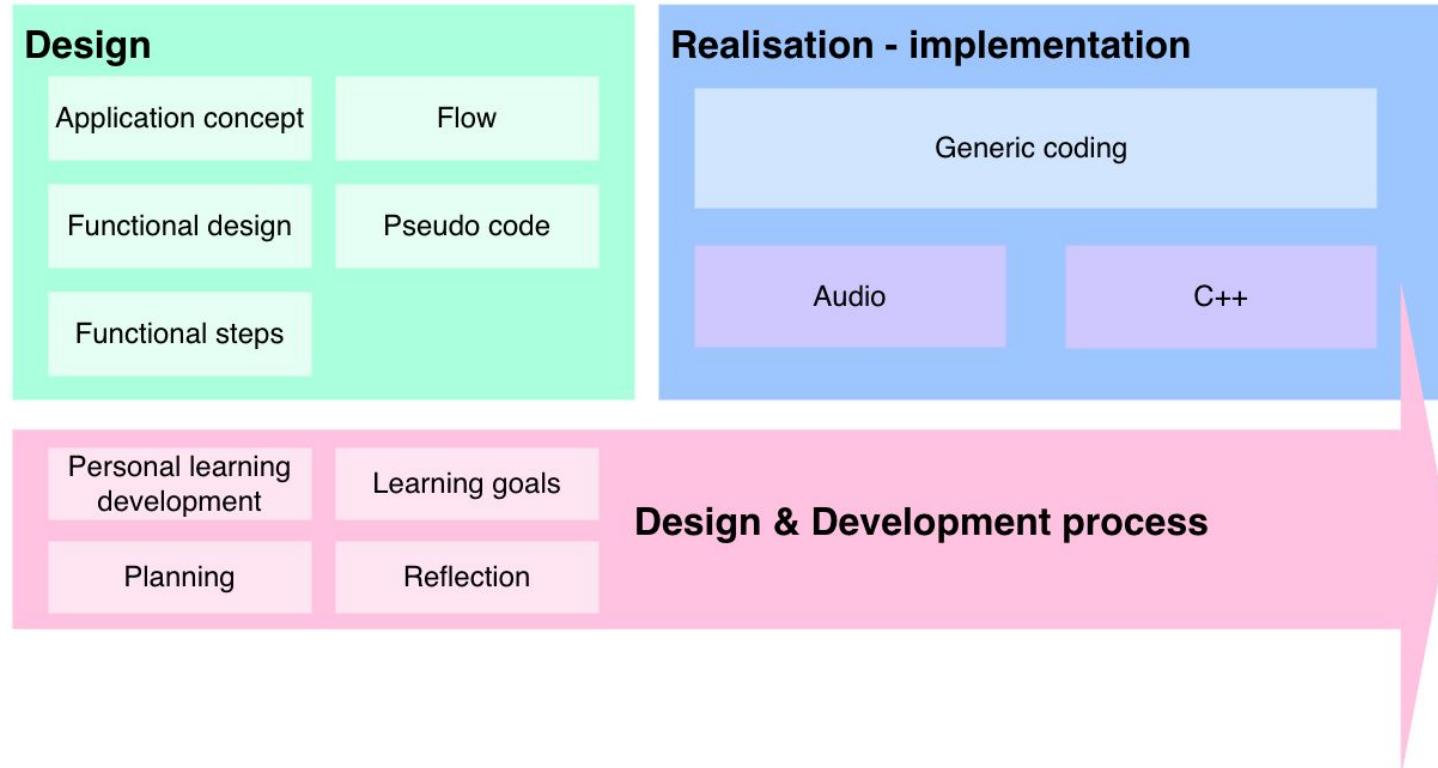
Uitgaande van 20 uur per week: (*6 ECTS voor 7 weken → 24 uur per week*)

- 6 contacturen per week
 - C++ theorie (Ciska, Daan) 3 klokuren + practicum
 - CSD Theorie (Pieter) - Signaal- en systeemtheorie II

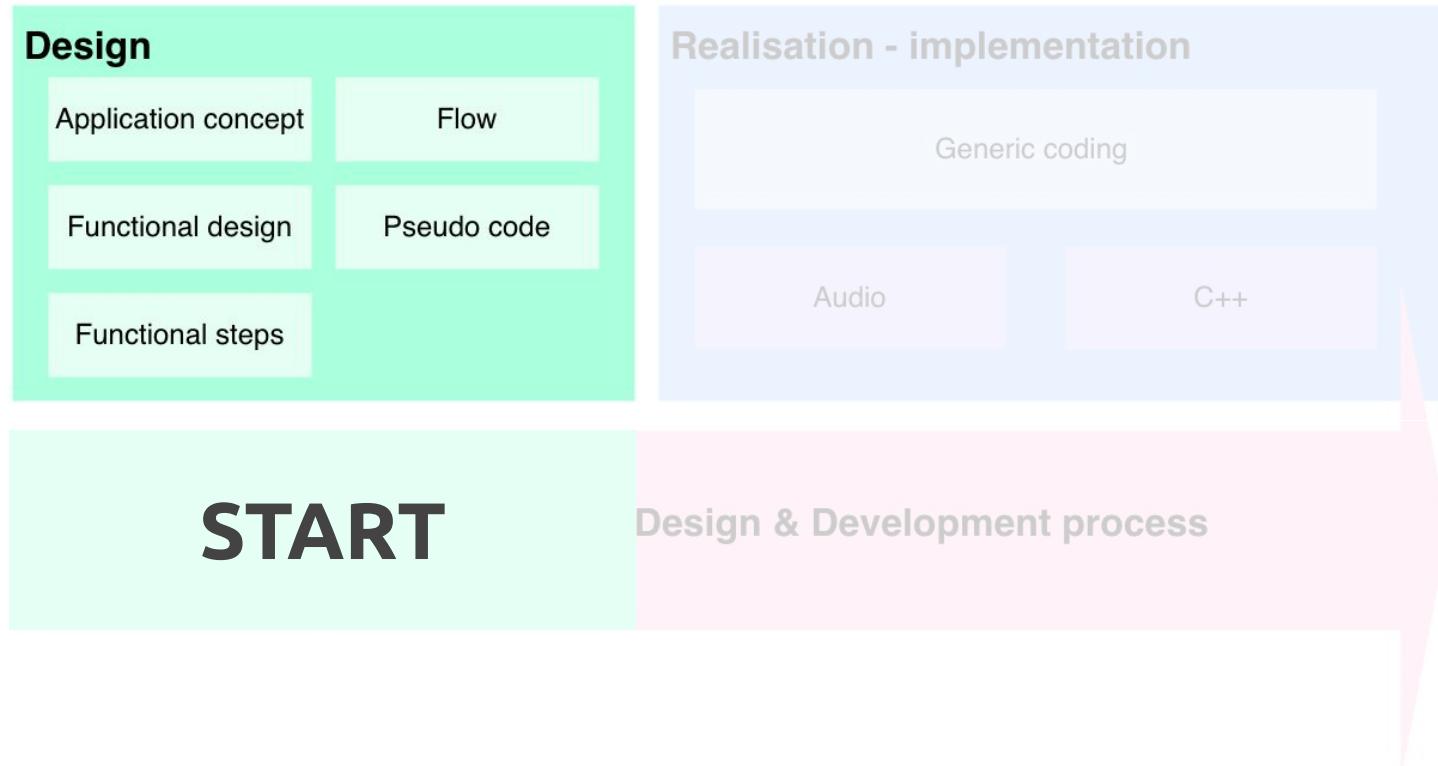
(Hardware ondersteuning (Hans - special basic) -- niet te volgen door roostering?)

- 14 uren zelfstudie

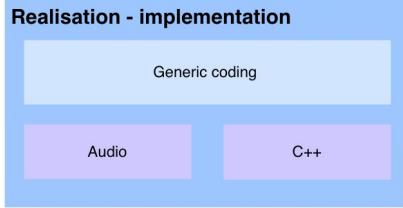
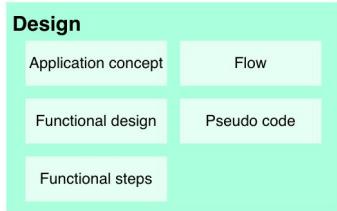
Opzet Blok CSD 2b



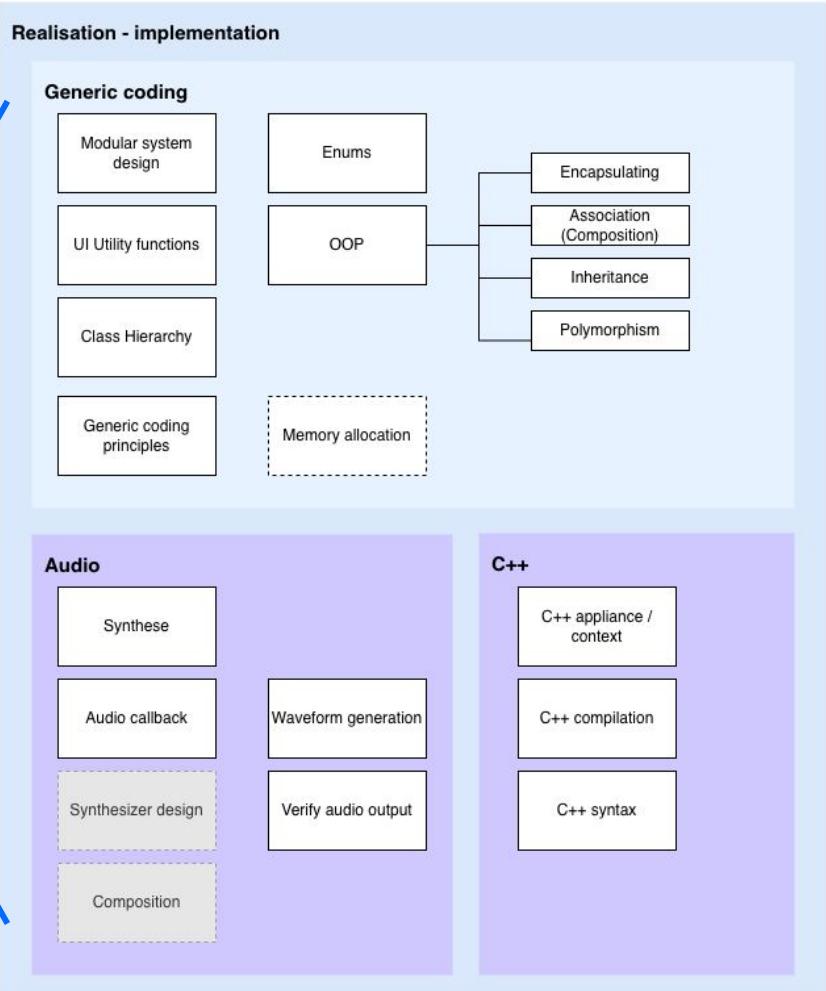
Opzet Blok CSD 2b



Opzet Blok CSD 2b



Design & Development process



Leerdoelen



Design & Development process

- **SYNTH FOCUS:** De CSD'er kan synthesizers ontwerpen en implementeren in C++ en hierbij de OOP principes correct toepassen.

OF

- **MELODY FOCUS:** De CSD'er kan een melody generatie strategy ontwerpen en implementeren in C++ en hierbij de OOP principes correct toepassen.

EN twee van onderstaande (*of één uit onderstaande én een goed geformuleerde eigen leerdoel*)

- De CSD'er kan een haalbare planning aanhouden met zelfstudietijd evenredig verspreid over de weken en kan hiermee een (ongezonde) eindsprint voorkomen.
- De CSD'er kan stapsgewijs aan afgebakende onderdelen van het ontwerp, systeem en implementatie werken.
- De CSD'er kan de time-management techniek timeboxing toepassen bij het onderzoeken / leren van nieuwe kennis en vaardigheden.
- De CSD'er kan wanneer zij/hen/hij vastloopt in het proces dit vastlopen herkennen en om hulp vragen.
- De CSD'er kan het beoogde ontwerp afstellen op de eigen capaciteit, zowel wbt het afbakenen ter voorkoming van overbelasting als opschalen ter voorkoming van onderbelasting.

Session content		Self-study	Deadline assignments
1	<ul style="list-style-type: none"> • Introduction • Classes in C++ • OOP-association & OOP-inheritance 	<ul style="list-style-type: none"> • <u>Design & goals</u> • Classes in C++ • Compiling with Cmake • JUCE up & running 	
2	<ul style="list-style-type: none"> • Peer reviews 'Design main assignment' • Sine function expressed in continuous, discrete time and as a function of phase (precision) • .cpp and .h files 	<ul style="list-style-type: none"> (• 'Redo' design & goals) • Inheritance in C++ • Sine and Square / Saw oscillators in C++ • Audio callback - 'plugin your sine' 	[S2] Design & goals
3	<ul style="list-style-type: none"> • Audio callback • Virtual and abstract methods • Oscillator base class, derived Sine and Square classes. • Note class & array • Class diagram 	<ul style="list-style-type: none"> • <u>Class diagram main assignment</u> • Oscillator base class, derived Sine and Square classes • Simple Synth in C++ • Note array 	([P3] 'Redo' design & goals)
4	<ul style="list-style-type: none"> • Playing a melody • UI • Pointers part 1 	<ul style="list-style-type: none"> (• 'Redo' class diagram main assignment) <u>Main assignment implementation</u> • SYNTH: base & derived classes • MELODY: melody generation • Pointers 	[S4] Class diagram
5	<ul style="list-style-type: none"> • Pointers part 2 • polymorfisme (• dynamic allocation) 	<u>Main assignment implementation</u>	([P5] 'Redo' class diagram)
6	<ul style="list-style-type: none"> • Good & bad practice examples • Questions & feedback 	<u>Finalize main assignment</u> (implementation, design documentation, class diagram and reflection documentation)	[End week 6] main assignment (• <i>design documentation</i> , • <i>class diagram</i> , • <i>codebase</i> <i>• reflection documentation</i>)
	KERSTVAKANTIE		
7	Eindpresentatie	Prepare presentation	[S7] Presentation

Overzicht

- Vragen: tijdens les en practicum
- Opdrachten en voorbeelden op Github en ook
 - Algemene Beoordelingscriteria
 - Herkansing CSD-opdrachten
- Bronnen voor zelfstudie:
 - [tutorialspoint cplusplus](#), [w3schools](#) en [cpreference](#)
 - “Designing Software Synthesizer Plugins in C++” van Will Pirkle

Assessment

Design

Application concept

Flow

Functional design

Pseudo code

Functional steps

Personal learning development

Learning goals

Planning

Reflection

Realisation - implementation

Generic coding

Audio

C++

Design & Development process

- Design documentation 25%
- Class diagram 25%
- Codebase 25%
- Reflection documentation 25%

Missing a deadline? → -5% per deadline?

Still subject to change

Deadline assignments

[S2] Design & goals

[S4] Class diagram

[End week 6] main assignment

- design documentation
- class diagram
- codebase
- reflection documentation

[S7] Presentation